SESSION

PROJECTS + SOFTWARE ENGINEERING + PROGRAMMING/LANGUAGE ISSUES

Chair(s)

TBA

Mobile Robotics as a Platform for Capstone Projects

Costa Gerousis, Anton Riedl, Dali Wang
Department of Physics, Computer Science and Engineering
Christopher Newport University, Newport News, Virginia 23606, USA

Abstract – *Several years ago we started the development of an autonomous mobile robotics platform as a theme for hands-on projects in our Computer Engineering and Computer Science programs. Since then our MObile Sensing and Exploration System (MOSES) has been the subject of nine independent study and capstone projects. The topics have ranged from purely software-oriented robotics simulation to hardware setup and microcontroller programming such as autonomous GPS-waypoint driving and obstacle avoidance. Overall, the initiative has been quite successful. Student response as well as faculty evaluations in general have been positive. As a consequence, we believe that mobile robotics is well suited as a theme for capstone projects at the undergraduate level. It allows students to work with complex equipment and to apply their programming and engineering knowledge, while still keeping the projects accessible and inexpensive.*

Keywords: Autonomous and Robotic systems, Science and Engineering education.

1 Background and Motivation

About four years ago, a decision was made in the Department of Physics, Computer Science and Engineering at Christopher Newport University (CNU) to develop an autonomous Mobile Sensing and Exploration System (MOSES) [1]. Apart from being of interest to several faculty members, the main reasons for choosing this theme of mobile robotics were its high attractiveness and also its relevance for our students. Several publications have demonstrated that frameworks with similar degrees of complexity could provide interesting student projects in computing related areas [2, 3, 4]. As such, we envisioned that the MOSES system would also become a platform for ongoing student projects.

For the initial planning of the system, following main criteria, which are being discussed below in more detail, were taken into consideration:

- the need for challenging, yet manageable projects
- ABET's requirements for engineering programs
- appropriateness for undergraduate and graduate students
- suitability for individual and team projects
- costs and sustainability.

Undergraduate students in our computer engineering (CE) program as well as in computer science (CS) and in physics are required to do a capstone project in their senior year. As a consequence, there is always a need for interesting, multifaceted projects, which are challenging, yet manageable within a limited amount of time. Given the varying level of expertise that students have acquired by the time they start their capstone, it is also necessary to offer projects with different degrees of complexity and difficulty. Mobile robotics provides this as it spans different areas of knowledge, which are in the realm of our department: robot control, communication and networking, real-time operating systems, high and low level programming, artificial intelligence and sensing.

In view of ABET accreditation of our CE program, CE students need to be able to work on projects that emphasize the engineering perspective and, thus, cannot be purely research-based. According to ABET [5], projects have to provide a major design experience with realistic constraints, in which students apply knowledge and skills acquired in earlier courses. Again, given the multitude of possible topics under the umbrella of mobile robotics, it is quite easy to cut out engineering projects.

Another requirement was that projects be appropriate for students at the undergraduate as well as graduate level. While most students at CNU are undergraduate students, our department offers a Masters program in Applied Computer Science and Physics, which also has a concentration in computer engineering. It is therefore important to be able to offer Master thesis topics with a focus on hardware and hardware-near software.

The platform has to be suitable for individual as well as team-oriented projects. Typically team projects are preferred as these contribute an important part to a student's educational experience [6]. However, the small size of our programs requires that students can work on projects individually in order to not delay their studies.

Finally, it was important that the individual system modules were affordable in order to allow students to buy equipment if they wanted to. Furthermore, this way the platform would be sustainable even through times with less available funding. Our decision with this regard was to build the system only with commercial off-the-shelf products (COTS), which would be added to the platform as needed.

Overall, the introduction of MOSES as a project theme has been successful. So far, nine students have been involved in the design and development of the system. Furthermore, we have seen positive results on the educational experience survey completed by graduating computer engineering students who have participated in the MOSES project. Such results showed higher ratings on the learning outcomes than for students who have worked on other, often self-proposed projects.

The remainder of the paper is organized as follows: We first describe the platform and its components, providing important design criteria. Then, three projects are presented in order to give an impression of the type of projects that are being carried out. Finally, initial observations and results regarding the educational outcomes are presented and analyzed.

2 The MOSES Platform

2.1 Drivetrain

As drivetrain we use a Traxxas Truck model E-Maxx, which has proven to be a solid and cost-effective construction. The mechanical platform of the robot was stripped down and its radio control unit bypassed in order to control the servos from a central drive control. The Traxxas model uses three separate servomotors for speed control, switching between gears and turning. All motors are controlled by Pulse Width Modulation (PWM).

In addition to this system, and mainly in cases when multiple students are working on the system concurrently, we are also making use of the DFRobot platform [7]. This is a 4-wheel-drive mobile system that can easily be paired with an Arduino microcontroller (see below). Both drivetrain models are illustrated in Figure 1.

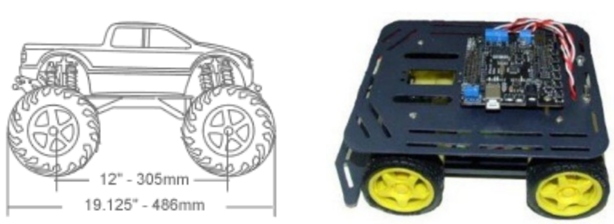


Figure 1: MOSES Drivetrain (Traxxas left, DFRobot right)

2.2 Robot Control

Originally, the design for MOSES included a Stargate processor board from Crossbow as the central processing unit of the robot. The Stargate provides a complete Linux machine based on a 400 MHz XScale processor. It includes a daughter card that adds Ethernet, USB Host, and serial connections. The main board has a Compact Flash (CF) and PCMCIA card slot. To release the on-board computer from the burden of frequent control action, an additional SSCII Mini motor control unit was employed. This microprocessor-based unit takes simple commands from its serial port and translates them into PWM signals. The programming of the Stargate-based drive control was done in two ways. The algorithms were implemented either in Java or through the robotics framework Player [8]. Player is an open-source robotics framework, which allows networked clients to drive a robot via a server running on the control board.

Several students worked with this system. However, the use of this software in combination with the Stargate board proved to be too complicated for undergraduate projects. As a consequence, it was decided to change the design and employ a simpler and better-supported architecture. We now use Arduino development boards as the basis for our robot control. Arduinos, which are built around Atmel microcontrollers, provide an inexpensive open-source prototyping platform. They offer many ways to interface with robotics hardware, including digital and analog I/O ports (some PWM capable) and various serial interfaces. Additionally, there is ample support for miscellaneous devices such as sensors, servos and communication modules. Most are already mounted on break-out boards compatible with the Arduino architecture. Programming is done using a Wiring-based language, which is very similar to C++.

2.2 Sensing and Wireless Communication

For the MOSES system to be able to act autonomously, various sensor capabilities are necessary. Furthermore, wireless communication functionalities have been implemented in order to allow the system to interact with human controllers or with other devices such as data collection stations or even additional autonomous vehicles.

Over time, following sensing and communication units have been added:

- A Global Positioning System (GPS) module, which is used for navigation capabilities. Communication between the microcontroller and the GPS module is implemented via the serial interface, where NMEA strings containing information on the speed, current time, and the current latitude and longitude coordinates are sent from the device to the controller.
- Two gyroscopes, which are being used to assist in navigation. In general, gyroscopes allow for precise measurements of the turn and tilt of the vehicle body. In our case, the two devices have different resolutions for the turning rate. One is used for determination of the general direction, while the second determines the direction more precisely during turning procedures. Integrating both gyroscopes enables the robot to keep track of large degree variances in movement while providing precision during turning procedures.
- An electronic compass for absolute heading calculations. This is especially important in situations where the vehicle is in standstill and, thus, the GPS cannot determine the heading.
- An ultrasonic distance sensor for obstacle detection and avoidance. The sensor is mounted on a 360 degree

rotation servo, which allows turning the sensor in any direction and enables MOSES to detect obstacles in its vicinity.

- A 433 MHz ultra low power data radio module for establishing one-to-one communication links over a relatively long distance. The selected modem has a range of about 1000 m with a bit rate of 1200 bps. This communication link is meant mainly to transmit simple control commands or heart beat signals between the robot and a control station.
- A 2.4GHz XBee module, which is based on the IEEE 802.15.4 protocol stack and which provides low-power communication over shorter distances (up to about 100 m). This form of communication is used for exchange of data between the robot and various base stations, which simulate data collection units. For future projects, it is planned to use this communication technology to establish an ad-hoc communication infrastructure between multiple MOSES systems.

As the platform has grown in functionality, the number of components has increased considerably and as a consequence a redesign of the overall hardware base has become necessary. The layout of the electronic components and the overall system platform are currently being optimized in a student project. The objective is to develop a cleaner structure, which supports rapid prototyping while offering a clear layout of the individual devices.

3 Selected Student Projects

In order to provide an impression of the type of projects students are working on within the MOSES framework, a few concrete examples are presented.

3.1 Stargate, Player and Stage

Two early projects were concerned with the implementation of the Player framework on the Crossbow Stargate device and with a simulation using the software Stage.

As mentioned above, Player is an open-source robotics framework developed by robotics researchers and used at labs around the world. The control system is centered on a server, which runs on the control board and interacts with sensors and actuators. The actual control algorithms are implemented in clients, which connect to the server via TCP/IP sockets.

The objective of the first capstone project was to get the Player server to work on the Stargate, which was running a version of Linux for embedded devices, and to demonstrate its functionality. This required the crosscompilation of the software for the appropriate architecture, the use of the serial interface in order to control the servos, as well as the implementation of simple clients to activate the servos. While this project was mainly software-oriented, it still required good understanding of hardware components and systems. As such it was very suitable for a computer engineering student.

The second project dealt with simulation of robots in Stage. One of the reasons for originally choosing the Stargate controller was its support by the Player software, which itself is closely related to Stage. Stage is a simulation environment that can be used in conjunction with Player in order to test client implementations. Stage provides a virtual world interface for Player, making it possible to develop and test control algorithms without having an actual robot.

We considered this ability to implement and test algorithms independently from the hardware a good way to also involve computer science students in the MOSES project. Unfortunately, the Player/Stage framework and its implementation on the Stargate architecture turned out to be too complicated to be applied effectively by our students. Therefore, we decided to switch to the Arduino controller architecture, which is not as sophisticated, but more intuitive and accessible.

3.2 GPS Way-Point Driving

This project was based on the Arduino platform. The robot is programmed to follow a path given by GPS way-points that are uploaded to the controller beforehand. The vehicle stops either when an emergency stop occurs or a waypoint is reached. An emergency stop is signaled by pressing a push button on the robot or by sending a particular command over the RF interface. Figure 2 illustrates a typical path.

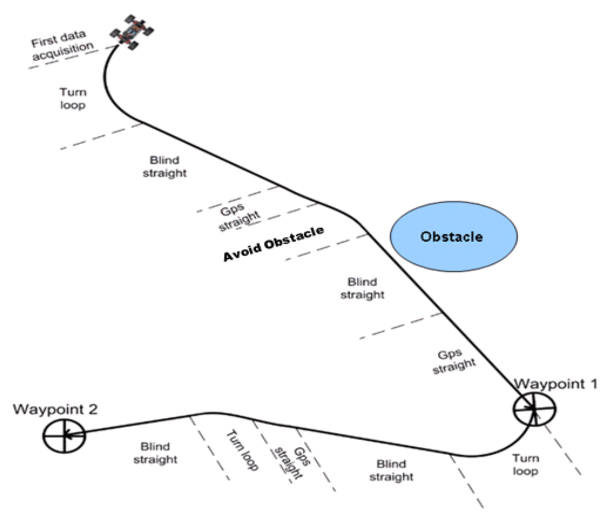


Figure 2: Typical path of the robotic vehicle

The driving algorithm is implemented as a state machine, which is indicated in Figure 3. This state machine demonstrates that while the Arduino platform is simple enough to be understood quickly by students at the undergraduate level, it still supports programs of considerable complexity. In this project, the Arduino controller drives two servos via PWM, reads in and processes the information from multiple sensors, and performs necessary control calculations, including a PID controller for turning and direction adjustment.

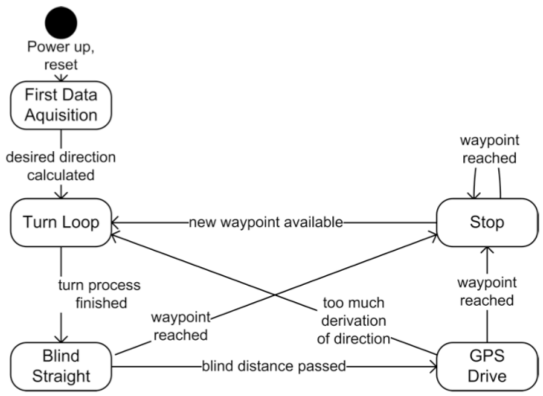


Figure 3: Way-Point State Machine

3.3 Obstacle Detection and Avoidance

Another project also based on the Arduino framework was concerned with obstacle detection and avoidance. For this, the smaller test robot was equipped with a ultrasonic distance sensor, a continuous rotation servo and the Arduino microcontroller. To allow for estimation of the size of an obstacle for proper avoidance, the ultrasonic sensor was placed on top of the servo. This allows for flexible readings of distances to objects within a certain range around the vehicle. Based on the readings the robot determines whether objects are in the way and if so, which direction it should take to avoid them.

The obstacle avoidance algorithm is based on a danger-grid used in Bayesian Occupancy Filtering [9]. During forward drive the servo stays positioned at a central point until it receives a reading of 4 meters or less. It then rotates the sensor in slight increments from left to right and reads in the respective distance values. The actual obstacle avoidance decision is based on intelligent Bayesian filtering where the individual sectors around the robot are assigned danger probabilities, i.e., probabilities that an obstacle is in the robot's path. Based on these probabilities, the robot makes a decision on which way to best drive around the obstacle.

Figure 4 illustrates the obstacle avoidance algorithm. At each point in time, there are basically four possible obstacle scenarios considered when making a decision on the path to take. The first is the simplest consideration, which is not having an obstacle at all. If no obstacle is detected, the vehicle continues on its current path. The second type of obstacle situation deals with an obstacle that is either dead center or too close to the robot. In this case, the robot will back up approximately a meter in distance and then it will check again for the dangerous paths ahead and avoid the obstacle. If there is most likely an obstacle located more to the left of the vehicle as determined by the Bayesian danger probability calculation, the vehicle makes a sharp right turn until the obstacle is out of view. The same happens if an obstacle is seen on the right side of the vehicle, except that the vehicle will turn sharp left until the obstacle is not in view and continue on its path to the waypoint.

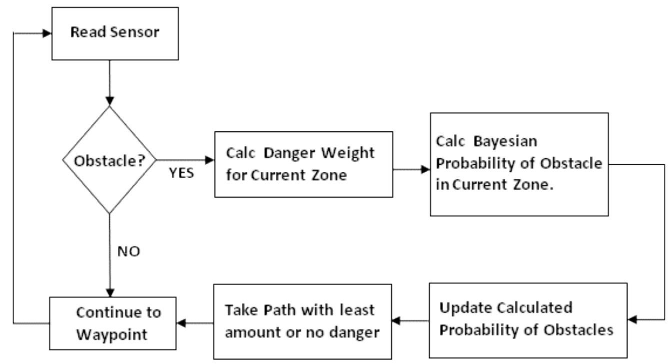


Figure 4: Obstacle avoidance algorithm.

4 Assessment of Educational Outcomes

As the majority of the students who have worked on MOSES so far were from the Computer Engineering program we focus our assessment on this group of students.

For the Computer Engineering program, the capstone course plays an essential role in two educational outcomes:

- an ability to design a hardware and software systems to meet desired needs
- an ability to use the techniques, skills, and engineering design automation tools necessary for effective computer engineering practice

As part of our program outcome assessment process, we conduct an exit interview with all graduating seniors who have just completed their capstone course. As part of the exit interview process, we ask our students to assess, on a scale from 1 to 5, the extent to which these program outcomes have been addressed. We compare the results from the students who have used MOSES as the main theme of their projects with the ones from all computer engineering majors over a period of 3 years. The summary of the student responses is presented in Table 1. The total number of students over this period who have worked with MOSES was 9. The number of students who have done a different project was 22.

Table 1: Comparison of educational outcome evaluation (student data)

Outcomes	Average (MOSES students)	Average (all students)
Design a hardware and software system	4.64	4.09
Use the techniques, skills, and engineering design automation tools	4.53	3.91

It is shown that the students in the MOSES group rated these two design-orientated program outcomes significantly higher than other students.

Faculty evaluation of the senior project is another part of our program outcome assessment process. Students completing these capstone projects are required to make a formal presentation before the faculty. At the presentation, faculty members fill in an evaluation form for each student project. Several aspects of student projects are evaluated, among them the two design-orientated program outcomes. The summary of the faculty responses is presented in Table 2.

Table 2: Comparison of educational outcome evaluation (faculty data)

Outcomes	Average (MOSES students)	Average (all students)
Design a hardware and software system	4.50	4.47
Use the techniques, skills, and engineering design automation tools	4.56	4.47

It is shown that faculty evaluation of the MOSES related project is only slightly higher than that of all projects. We think that two factors contribute this phenomenon. First, students tend to demonstrate more personal interest in the MOSES related projects (the "cool" factor), while faculty tend to look into each work more objectively. Second, all faculty have been actively involved in mentoring student capstone projects. The department as well as the university has a long history of valuing faculty mentoring of student research. As a result, most capstones are well-defined engineering projects targeting our computer engineering majors.

5 Conclusion

Overall, the MOSES system has met many of the expectations that were initially put into this platform. It has proven itself to be very suitable for undergraduate capstone projects, specifically for computer engineers. As the sample projects demonstrate, MOSES provides many opportunities for appropriately complex capstone projects where students get to design and implement solutions to typical engineering problems. Through MOSES students have been exposed to a number of technologies, hardware as well as software, and they have had the opportunity to integrate various modules into a coherent system and write the programs for this.

With regards to educational outcomes, our initial assessment indicates that students typically find the work with MOSES interesting and also educational. Most importantly, they feel they have gained experience in two areas that are central to the engineering education: in the design of a hardware and software system, and in the use of techniques, skills, and engineering design automation tools. The average ratings they give for both of these outcomes are higher than the ratings given by students who were working on other projects. When it comes to faculty evaluating the projects, this advantage is not as clear-cut. However, this discrepancy seems to be due to a more objective and more balanced evaluation of projects by the faculty members. In their view, all projects receive equally high values.

6 References

[1] A. Riedl, D. Wang, C. Gerousis, R. A. Flores & David Doughty, "A Robotics Platform for Capstone Projects in Computer Engineering and Computer Science," 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS '09), Las Vegas, July 13-16, 2009, pp. 217-222.

[2] J. J. Evans. "Undergraduate research experiences with wireless sensor networks." Proceedings of the 37th Frontiers in Education Conference, pp. S4B-7 - S4B-12, October 2007.

[3] A. Saad1, "Mobile Robotics as the Platform for Undergraduate Capstone Electrical And Computer Engineering Design Projects." 34th ASEE/IEEE Frontiers in Education Conference, pp. S4B-7 - S4B-9, October 2004.

[4] J.K. Archibald & R.W. Beard. "Goal! Robot soccer for undergraduate students." IEEE Robotics & Automation Magazine, Volume: 11, Issue: 1, pp. 70 – 75, March 2004.

[5] ABET. "Criteria for accrediting engineering programs (2008-2009)", ABET Engineering Accreditation Commission, 2008.

[6] L.R. Lattuca, P.T. Terenzini, & J. Fredricks Volkwein. "Engineering Change – A study of the impact of EC2008". Joined published by ABET and Center for the Study of Higher Education, The Pennsylvania State University, 2006.

[7] DFRobot. Internet: http://www.dfrobot.com/index.php?route=product/product&product_id=116

[8] R.T. Vaughan & V.P.Gerkey "Reusable Robot Software and the Player/Stage Project." D. Brugali (Ed.), Software Engineering for Experimental Robotics 2007, pp. 267 – 289, Springer-Verlag.

[9] H. Hu and M. Brady, "A Bayesian Approach to Real-Time Obstacle Avoidance for a Mobile Robot," Autonomous Robots, 69-92, 1994.

Third-Year Parallel Programming for CS Undergraduates

William B. Gardner
School of Computer Science, University of Guelph, Guelph, ON, Canada

Abstract - *This paper describes a successful new course aimed at helping soon-to-graduate students move into jobs using current tools for parallel programming, by acquiring the theoretical background needed to keep abreast with rapid industry developments and to evolve with them. It intentionally spans the range of multicore to cluster computing, based on the same underlying principles. All aspects of the course are described, including textbook, schedule, lab content, assignments, projects, and outcomes after two offerings.*

Keywords: Parallel programming, Undergraduate CS curriculum, Multicore, High-performance computing

"I help search for water on Mars." At the first meeting of our new Parallel Programming course, rolled out experimentally as a Special Topic offering in Fall of 2009, we were going around the room, each student invited to say why he or she had registered and what they hoped to learn. Among the expected third- and fourth-year CS undergraduates was this one graduate student from the Physics Department. Her research group analyzed spectrographic data radioed back from the Mars Rover for telltale signs that water molecules could be present. In order to interpret the spectrograms, they would compare them to those of simulated bombardments by 50 billion photons onto various artificial compositions of Martian "soil" spiked with specific amounts of water. Each run for a single soil/water composition would take 24 hours to complete on a high-end desktop computer, and she was the only person in the lab who considered this state of affairs decidedly subpar for the new millennium of computing!

For CS students who were used to coding assignments that instantly compiled upon pressing Enter, and executed in tens of seconds at most, this was a revelation. In the event, she lacked the programming background to stay with us, but her case study provided invaluable motivation for launching the course.

1 Background and introduction

Our School offers a Bachelor of Computing degree with two majors, classical computer science and software engineering. For the last several years, it was obvious that we would have to come to grips with teaching parallel programming beyond the basic introduction to concurrency that has long formed a topic in Operating Systems (OS). As the faculty member whose research has centered on formal methods for specifying concurrent systems [1], with background in digital hardware as well as software engineering, I was relatively suited to the task of examining our options and creating a suitable course.

Specifying the OS course as a prerequisite, the 2009 Special Topic attracted 17 students. It was repeated, again as a Special Topic, in Fall 2010, with modifications based on the first experience. This time 26 students signed up, including another graduate student in Physics (this one stayed and did an impressive project). Based on student surveys, the course can clearly be called a success. By Fall of 2011, it will enter the regular curriculum as CIS*3090 Parallel Programming, an annually offered elective.

This paper covers every important aspect of the course, for the benefit of those who may wish to copy or adapt it, including the following:

- Approach: why we chose to mount a single special course, as opposed to incorporating parallel programming topics into existing core courses
- Characteristics: breadth over depth, wide-spectrum (multicore to high-performance clusters), theory-based with programming practice, tools-oriented (for profiling and error detection), programming techniques selected for prospects of longevity, special-hardware focus avoided, assumed prerequisites, target audience
- Organization: lecture topics, 12-week schedule, programming platforms, software tools

- Textbook: strengths and weaknesses
- Assignments: rationale, coordinated hands-on lab sessions, contests
- Project component: categories, sample of topics, approach to evaluation
- Human resources: instructor workload, use of teaching assistants

2 Approach

From the standpoint of impact on curriculum, the first question to settle is whether a CS department will attempt to integrate parallel programming concepts into several existing core courses, or will mount a standalone course, or even do both. With an eye to the burgeoning multicore future, it seems wise to start “parallel consciousness raising” early, and we do not oppose that approach. Nonetheless, pursuing some degree of integration does not detract from the benefits of mounting a full course—in particular, training students to have job-ready skills—and indeed could allow that course to have a more advanced starting point. Furthermore, in a department such as ours, there are enough drawbacks and stumbling blocks so as to make it a choice between mounting a standalone course right away, versus doing nothing helpful for possibly a long time. In the end, we took the standalone approach for several reasons:

First, it was less disruptive to the faculty and curriculum as a whole. The core courses in our 12-week semesters are already “full,” so that adding topics necessarily means displacing something else, which, if shifted around, can widen the impact to other courses.

Second, while it has been suggested that we should cease teaching sequential programming and just carry on with parallel programming [2], this is not convincing. Parallel programs are made of sequential portions, and applications that don’t require special performance will continue to be written adequately as sequential codes. Similarly, going back to the early prediction that object-oriented programming (OOP) would displace procedural programming, this has not occurred, and we see that OO methods are made up of procedural code. Like many departments, we continue to teach both styles of programming with full courses, and there is no good reason to think that sequential programming courses can simply be converted to parallel courses. For a number of years to come, it will be necessary to take up the latter as additional subject matter.

Third, there is the problem of personnel. Given that most present faculty members were educated in the sequential programming era, how are they going to readily teach parallel topics? It is more straightforward to concentrate the responsibility for that new knowledge and skills in some volunteers who already have suitable background, and let them create a course.

Finally, model curricula with integrated parallel topics are only now being proposed (for example, see [3]), and being an early adopter is risky. Many departments will prefer to learn from others’ experiences before tampering with their own core courses, which may be based on textbooks that do not have parallel topics. In contrast, the investment in a standalone course can pay benefits right away in terms of student satisfaction. And if the department eventually moves to introduce parallel topics early, the programming course can adapt by starting from a more advanced level, which is a win-win outcome.

3 Characteristics

What is the target audience of students, and which prerequisites should be demanded? We decided to aim for third- and fourth-year students, for whom programming per se is no longer a big challenge, thus they are free to concentrate on grasping new concepts and applying them with the programming skills they already possess. Two essential elements of background knowledge are concurrency—which in many OS courses is taught along with POSIX threads, and exposes students to critical sections, deadlocks, resource contention, mutexes, and condition variables—and basic computer architecture. For decades, computer and OS designers have dedicated themselves to making the hardware largely invisible to software, and may have succeeded too well. By now, we find that many students will only take an architecture course grudgingly, and feel there is no purpose in it. And yet, in learning how to get maximum speedups out of parallel programs, it is necessary to draw in issues like memory bandwidth limitations, cache coherency, and false sharing. If students are going to understand why a program’s speed may depend on the arrangement of a data structure in memory, they have to know something about hardware. We find that our second-year Structure and Application of Microcomputers course gives enough exposure that they can follow the hardware issues related to parallel performance.

When parallel programming courses started to be mounted in universities, a typical approach was to focus on a certain architecture, e.g., IBM Cell BE or GPU. But, emphasizing special hardware carries risks of rapid platform evolution or obsolescence. We prefer to give students a broad introduction ranging from the now-ubiquitous multicore desktop to the previously established world of high-performance computing (HPC) clusters. These disparate platforms have both similarities and differences that help students understand the underlying hardware issues, and the advantages and limitations of various programming techniques. We chose to teach POSIX threads, OpenMP, and message-passing programming because they leverage, to some extent, concurrency background taught in the OS course, are widely practiced in industry and academia, and show no sign of going away.

4 Organization

Organizing any CS course is a major undertaking that necessitates consideration of textbooks or other learning resources, programming languages and computing platforms, often supplied via in-house computer labs, additional software packages and training in their use, plus plans for assignments, exams, and possibly term projects.

The various components of the course were weighted into the grade as follows:

- Assignments 30%
- Term project 35%
- Final exam 25%
- Participation 10%

The exam was based on the textbook and the three programming methodologies. Participation marks were given to encourage attendance at, and peer evaluation of, the project presentations.

Since the textbook is a kind of linchpin, the subsections below start with that, and then the schedule of lecture topics is presented, coordinated with textbook chapters. Next, the programming platforms and software packages are described. The plan for the programming assignments is given, and then for the projects. Finally, utilization of human resources is described.

4.1 Textbook

In 2008 and 2009, there were as yet few solid entries into the market for parallel programming textbooks. Fortunately, I have been very content to discover *Principles of Parallel Programming,* by Calvin Lin and Larry Snyder, Addison-Wesley, 2009. It strikes the right balance for a university setting, between mastering techniques and tools, on the one hand, and presenting a theoretical basis, on the other. The authors introduce their parallel pseudocode, called Peril-L, which is suitable for implementing as pthreads, OpenMP, or message-passing. Similarly, the theory component is equally applicable across the spectrum of parallel platforms. Sufficient hardware description is supplied to explain phenomena that must be grasped in order to produce scalable programs.

The first chapter is captivating: It commences straightaway with a simple case study that points out several common pitfalls in parallel programming, e.g., the parallel version runs slower than the serial version, race conditions produce incorrect results, it is not very scalable with more cores, and so on. From a student's standpoint, this immediately raises the stakes from "here we are, learning yet another programming language (which I could have taught myself)" to "maybe there is something I don't know after all!" This has the effect of strongly motivating the course, and it plays into an important theme: the *computer professional* knows how to obtain good results (here, parallel performance) by applying knowledge and skills; the *hacker* gets good results, if at all, mainly by luck.

The main weakness of the first edition—possibly a symptom of being rushed into a hot market—is a large number of errata, most of which are noted on the authors' website [4]. One can only hope that a second edition will be printed to solve these problems.

Two other books were put on the course syllabus as recommended reading: *Patterns for Parallel Programming,* by Mattson, Sanders, and Massingill, Addison-Wesley, 2005; and *The Art of Multiprocessor Programming,* by Herlihy and Shavit, Morgan Kaufmann, 2008.

The Lin and Snyder textbook is organized into sections. We studied the entire first section's five chapters, which provide the necessary conceptual basis for program development. Specific programming methodologies are covered in chapters 6 to 8, classified as threads, "local view" languages, and "global view" languages, respectively. One is free to pick and choose among them. Chapter 9 gives an assessment of existing approaches, and becomes rather abstract for our purposes. Chapter 10 surveys "future directions" and goes well alongside overviews of selected platforms I chose to introduce: NVDIA GPU with the CUDA language, OO threading libraries from .NET and Intel, and the IBM Cell BE. Anticipating use in a project context, the book ends with a practical "capstone project" chapter (11).

The overall timing strategy for the 12-week course involved three stages:

1. Laying the conceptual groundwork for parallel programming, based on chapters 1-5.
2. Learning three specific programming methodologies applicable to non-shared memory (cluster) and shared memory (multicore) platforms, with one assignment each.
3. Surveying a variety of topics while students were carrying out projects utilizing the above methodologies. If experts on specialized topics are available, this is an ideal time to bring in guest lecturers.

The schedule of topics, coordinated with textbook chapters and labs, is shown in Table 1. There was a definite purpose in the order of teaching the programming methodologies:

Table 1. Course schedule

Unit	Reading	Lab
1. Introduction	chs. 1 & 11 Getting Started	
2. Understanding Parallel Computers	ch. 2	
3. Reasoning about Performance	ch. 3 to Trade-Offs	Pilot library
4. First Steps Toward Parallel Programming	ch. 4	
5. Scalable Algorithmic Techniques	ch. 5	Intel Parallel Studio
6. Programming with Threads • POSIX Threads • OpenMP	ch. 6 (parts)	
7. Preparing for Project	ch. 11 Capstone	
8. Assessing the State of the Art	chs. 9, 3 & 11 (finish)	
9. Future Directions in Parallel Programming	ch. 10	

The first one, message-passing programming for high-performance clusters, utilizes an in-house library, Pilot [5], which is a simple process/channel abstraction layered on top of conventional MPI, and targeted at novice scientific programmers. Pilot's process definitions are similar to pthread_create (which students know from the OS course), and its distinctive C API is modeled on stdio's well-known fprintf/fscanf, so it is simple to teach and more difficult to abuse than MPI. Pilot also includes an integrated deadlock checker that is capable of diagnosing right to the line number in program code that misused of the API or caused a circular wait, for example, thus preventing the phenomenon of silent, hung programs commonly experienced by beginning MPI users.

The above features make Pilot very suitable to teach while the course content is just starting to build, and students can use it for the first assignment. Those who wish to do a cluster-based project on hundreds of processors can still use Pilot (which also has a Fortran API), or they can branch out and learn the more complex, low-level MPI, for which Pilot will have given them good preparation. Pilot is available for free downloading from its website [6], and installs with any MPI.

Next comes pthreads with Intel Parallel Studio tool support; and OpenMP, also with Parallel Studio. Pthreads is taught before OpenMP, both to connect back to students' OS course experience, and because OpenMP is deceptively easy to use, to the point of making them feel that Pthreads programming is too onerous. Assignments #2 and #3 involve implementing the same program in pthreads and OpenMP, which helps them to closely compare these technologies. The purpose of using Intel Parallel Studio (available under free academic license) is to employ its tools Parallel Inspector, for detecting shared memory conflicts and potential race conditions, and Parallel Amplifier, for profiling program performance down to the core level. These tools strongly support the common use case of parallelizing existing sequential programs. Students find it very illuminating to see just how much of the time their program spends utilizing n cores (where n=1-16 on our system).

4.2 Parallel programming platforms

In order to offer an exciting suite of high-end parallel hardware for assignments and projects, we were able to procure a "pre-owned" 32-node Itanium cluster from SHARCNET—our university's associated high-performance computing consortium—running Linux, and a 16-thread Mac Pro Core i7 running Windows Server under Apple's Bootcamp. (SHARCNET accounts were available for projects wanting hundreds of processors.) This enabled students to obtain hands-on experience typical of both the HPC world and systems used in industry. The same Intel C/C++ compiler was installed on both systems.

Intel offers a free Summer School with training on Parallel Studio. It is that training material that we use successfully for the course labs. The Pilot hands-on labs are the same half-day tutorial that we run at international conferences. Those labs are also available for downloading from the Pilot website.

4.3 Assignments

The students carry out one assignment using each of the three programming techniques, starting with Pilot. As stated above, the pthreads assignment gets rewritten using OpenMP for the purpose of comparing and contrasting.

A key requirement for all assignments is short written reports describing the rationale for the student's parallel design, and featuring timing, speedup, and efficiency graphs against an X axis of number of processors, all with the student's interpretations. Timing is done both with and without compiler optimization. Students are often shocked to see that the compiler can cut execution time by as much as one-third. They must provide proof of program correctness, and explanations for what they learned from refining their programs to improve performance. These write-ups showed that they were able to understand and apply the theory we had learned.

The three graphs basically show the same data in different ways, yet they are not redundant and one can learn something more from each of them. The timing graph (wall clock time) gets right to the obvious objective: how fast is my program? The speedup graph, showing the ratio of serial time vs. parallel time, serves to factor out compiler optimization effects, and yields a frank assessment of scalability or lack thereof. The efficiency graph, speedup divided by number of processors, shows how far their parallel performance falls short of the perfect "1.0" efficiency line.

Bonus points were awarded for the fastest solutions, in keeping with the parallel performance emphasis. Those finishing in second and third place were allowed to challenge the winner to a rematch, which gave them the experience of consciously trying to tune their programs, and often the ranking—and the bonus points—changed hands.

4.4 Projects

As important as programming assignments are, they have to be tightly specified and of limited scope. To really apply what they are learning, doing an independent project is invaluable. Students were allowed to work as individuals or form teams of two.

Each student or team has to propose (and get approved), program, and present to the class a project chosen from five categories, some of which come from the textbook. A written project report is handed in along with their source code. The categories, with samples from the two years, are as follows:

1. (Re)implement existing parallel algorithms: checker playing, cryptology, Quine-McCluskey method
2. Compete with standard benchmarks (no one chose this)
3. Develop new parallel computations: force histograms for 3D vector images, traffic simulator, gaussian blur
4. Pilot-related development: porting Pilot to Lua (LuaPilot) and Python (Pylot)
5. Exploring beyond the course, which could involve parallel languages that we did not study as a class: F# (solving Minesweeper), C# TPL (tree search) Cilk++, CUDA (particle simulation), OpenCL (image processing)

The approval step is advisable for ensuring that students do not recklessly launch into unsuitable topics, and that the scope is compatible with the few weeks left in the course, or that they at least have a fallback strategy in case things go more slowly than they expect. There was no formal deadline for the proposal, which was intended to facilitate rapid turn-around by avoiding having them all submitted at once. Counterbalancing that flexibility, students were required to provide weekly progress logs of their project activities—they could read each other's logs—and points were deducted if the instructor's weekly spot check found no update. This served to keep them moving along.

This "exploring" feature is an excellent way to keep the course up to date, by encouraging students to try out the very latest technologies and report to the class, which in turn enables the instructor to freshen the course without having to continually revise the instructional components.

Because of the knowledge they have gained, it is typically straightforward for the students to relate most tools or techniques to concepts they have learned, comparing and contrasting with methodologies they all (now) understand and have experience with. This gives the students a lot of confidence when they see they are capable of understanding and trying out some new parallel programming technology on their own.

4.5 Human resources

In terms of instructor workload, the projects require the greatest time commitment, depending on how thorough one wishes to make the evaluation. I found the projects to be so interesting, that I wanted to build and run each one. This is probably the least scalable component of the course as it is currently constructed.

If teaching assistants are available, one good place to use them is for "hand-holding" in the lab sessions, that is, walking around and assisting anyone who is having problems doing the exercises. Another worthwhile use is for compiling and running all the submitted assignments, checking for correct output, and making timings for the contests. This leaves the instructor to read and evaluate the paper reports. Someone who took the course before makes an ideal TA, otherwise it would be difficult to find anyone qualified who has only come through standard CS training.

5 Conclusion

Today's computer science students are entering a new era in parallel computing, featuring cheap multicores and high-performance clusters, but have received traditional largely-sequential training. Based on our experience with this course, we found that resources are presently available to mount a standalone course relatively cheaply. It meets CS students' practical educational needs, and it can be extremely gratifying to teach.

One measure of success is how the students view themselves as parallel programmers. After each course, they were asked to fill in an anonymous survey including the statement "I think I can handle parallel programming" (0-10 scale), comparing before vs. after the course. Their confidence rose impressively, on average, from 2.8 to 9.2. That confidence, plus all the languages and tools to write on their resumes, should give them a significant employment advantage.

If, in the future, our department decides to begin integrating parallel programming topics into the existing core courses, that will simply strengthen students' preparation for this dedicated course.

6 References

[1] William B. Gardner. Converging CSP specifications and C++ programming via selective formalism. *ACM Trans. on Embedded Computing Sys.*, 4(2):302–330, 2005.

[2] Wen-Mei Hwu, David Kirk, Christoph Lameter, Charlie Peck, and Michael Wrinn. There is no more sequential programming. Why are we still teaching it? In *Super Computing (SC08)*, Education Program, Panel Discussion, Austin, TX, Nov. 17 2008.

[3] NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing: Core topics for undergraduates [online]. Available from: http://www.cs.gsu.edu/ tcpp/ curriculum/index.php.

[4] Errata for Principles of Parallel Programming [online]. Available from: http://www.cs.utexas.edu/ lin/ errata.html.

[5] John Carter, W.B. Gardner, and G. Grewal. The Pilot approach to cluster programming in C. In *Proc. of the 24th IEEE International Parallel & Distributed Processing Symposium, Workshops and Phd Forum*, Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC-10), pages 1–8, Atlanta, Apr. 23 2010.

[6] Pilot home [online]. Available from: http:// carmel.socs.uoguelph.ca/pilot.

Incorporating Emerging Technologies and Lifelong Learning into Capstone Projects

L. Donnell Payne
Computer Science Department, Texas Christian University, Fort Worth, Texas, USA

Abstract – *Numerous computer science educators have reported on characteristics of capstone projects. Typical features include the use of appropriate software development process models, team collaboration, planning and management activities, demonstration of verbal and writing skills, reviews and critical evaluation, the opportunity to work on real world problems, and the incorporation of new concepts and technologies.*

The author presents two exemplary capstone projects that included a significant learning component and discusses the relevance of this requirement to project outcomes. Results from these capstone projects suggest that incorporating technologies, new to students, significantly improves student interest and engagement as well as providing students with the confidence and appreciation for lifelong learning.

Keywords: Capstone course, lifelong learning, GIS, gesture recognition, mobile applications

1 Introduction

1.1 Characteristics of capstone projects

Computer science educators have reported on numerous attributes and requirements of computer science capstone projects. Of significant importance in the capstone project is that it "cap off" the undergraduate experience by requiring students to apply skills and knowledge obtained in the program to produce an exemplary senior project.

Representative capstone course projects are characterized by:

- an extensive requirements gathering, design, implementation, and testing effort (generally a two-course, year-long project) utilizing an appropriate software development process;
- project development in a team environment demonstrating the development of interpersonal skills;
- use of project planning and management skills;
- demonstration of oral and written communication skills through numerous project presentations and documentation requirements;
- several reviews and critical evaluations – of both the product and the team members;
- selection of projects that have applicability outside the classroom, often with external sponsors; and
- incorporation of a significant learning component required for the project.

Most capstone experiences reported in the literature have the majority of these requirements [2, 3, 5, 6]. Many differ on the emphasis or type of projects developed. For example, some place more emphasis on creating real world solutions [1, 5] whereas others believe that learning the process is more important than the actual product [8].

1.2 Capstone course at this university

The capstone course for computer science majors at this university is a challenging two-course sequence. The first semester is primarily a lecture-based Software Engineering course but also includes preliminary project work. By the completion of the fall semester, projects and teams have been determined, requirements elicited, and an initial requirements specification and design produced. The second semester is project-based and includes requirements revisions, additional design work, implementation, testing, and the deployment phases of the project.

The course format and requirements are not unlike those of other programs. Students are required to select from candidate projects and work in teams. They are required to plan and manage their own project, post weekly activity reports, and provide periodic peer evaluations. Each team must specify, design, implement, test, and deploy a complex software system using an appropriate software development process. Version control software is used in all projects. All teams deliver appropriate documentation, with revisions, and have numerous oral and written communication requirements.

Several report increased student interest and motivation in projects developed for external clients and projects exploring new technologies [5, 6]. At this university, considerable effort is given to identifying projects that require the students to explore new technologies and projects that have applicability outside the classroom. When possible, it is also desirable to involve interaction with outside clients.

1.3 Overview of paper

This paper describes exemplary capstone projects of two recent years that required extensive student learning of both the problem domain and of technologies new to the students. Further, it relates the author's belief in the importance of incorporating significant student learning in the capstone projects in order to maintain student interest and motivation in the intense, year-long effort.

The remainder of this paper is organized as follows. Section 2 introduces a mobile geographic information system (GIS) developed for mapping and data collection on mobile devices in the field. It was originally developed to give forestry consultants the ability to map and obtain forestry data while on location. Section 3 discusses an acceleration-based gesture recognition system that was developed as last year's senior capstone project. The intent of the project was to investigate the use of mobile devices, with acceleration-based motion sensing capabilities, as innovative interaction devices. Section 4 summarizes project results and observations. Section 5 provides some concluding remarks.

2 Mobile GIS project

2.1 Background

The mobile GIS project originated from the need of forestry management personnel to obtain and utilize geo-referenced timberland information in the field. While it is true that there is at least one commercial GIS application that could be adapted for this application [4], that system requires the user to have extensive GIS training and, when integrated with the desktop components for display and analysis, is quite costly.

Forestry management consultants typically divide large timberlands into multiple subplots for management purposes. They maintain maps and management histories on each timber subplot comprising the entire tree farm. This history includes such things as harvest dates, harvest amount, undergrowth burn dates, acreage of each subplot, and estimated timber on each subplot. The on-site creation of custom maps and collection of associated data with editing and presentation capability was needed.

The goal of this project was to produce a low-cost, general-purpose mobile GIS application that provides for mapping and data collection in the field, is easy to use, and could readily be applied to this and many similar domains.

2.2 Learning new technologies

The system was developed for Windows Mobile touchscreen devices using Visual Studio and C#. Windows Mobile 5 and 6 emulators were used extensively in the development phase. Also, stand-alone GPS receivers with Bluetooth (BT) connectivity were selected for project development because they allowed field testing of all features of the system on each of our test platforms: laptop (utilizing the emulators); the HP iPAQ 2970 Pocket PC; and the AT&T 8925 Tilt smartphone. The i.Trek M5+ BT GPS receiver was selected for development and performed well. All of these technologies were new to the students as they had not developed in the .NET environment or worked with GPS prior to starting the project.

In addition to learning how to acquire GPS data through BT connections, the students had to learn fundamentals of mapping, map projections, and geographic information systems. This included learning how to obtain latitude and longitude information from NMEA satellite data and transforming and projecting geographic information from the 3D earth onto a 2D screen. This was not a trivial task.

2.3 System functions

The mobile GIS system developed provides four basic functions: Build Map, Edit Map Data, Map View, and Remote Tracking. Each is briefly described below in order to give an appreciation for the magnitude of the project.

The Build Map and Remote Tracking functions require latitude and longitude values to be obtained in real-time from an active GPS receiver. Figure 1 shows the main menu and the GPS status screen with the GPS currently active and tracking 6 satellites. These screen shots were obtained from the emulator running on a laptop with the Bluetooth GPS device attached.

Figure 1. Main menu (left) and GPS status (right).

The *Build Map* function allows the user to construct a map using any combination of 4 sampling methods. Map objects may be created by simply touching the screen with the stylus, by manually entering latitude and longitude values, by collecting arbitrary GPS points, or by sampling GPS values at a user selectable time interval.

Edit Map Data provides the ability to enter and obtain information on map objects. Properties of map objects are determined by a user defined XML configuration file. Figure 2 shows a map with pine timber subplots (dark polygons) on the lower right side. Timber subplot 108 has been selected and its associated properties (acreage, volume of standing timber, last cut year, and last burn year) displayed in the screen on the right. The map was built using several of the map construction methods previously noted. This included manual entry of latitude and longitude values for perimeter corners, use of the stylus for placing some of the cross-fencing, and real-time GPS interval sampling for mapping the creeks, roads, and subplots.

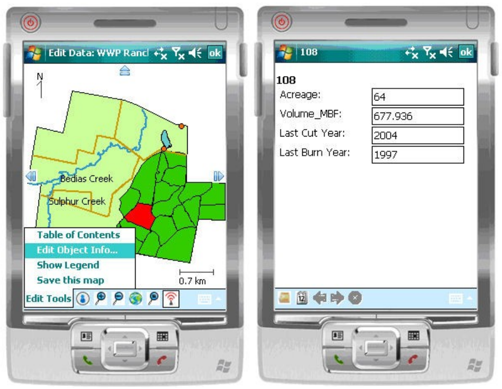


Figure 2. Edit Map Data with subplot selected (left) and data for selected object displayed (right).

Map View provides the user with common interactive techniques such as selection, zooming, panning, and "center on me". The map objects created are defined in the Table of Contents and placed on layers. This gives the ability to show/hide layers, modify layer ordering and label with selected properties. Map View also provides the ability to form queries and have selected objects highlighted. For example, Figure 3 illustrates a query for all timberstands greater than 40 acres that were harvested in 2004. Three subplots are shown selected in the screen on the right.

Figure 3. User query with selected Timberstands highlighted.

And finally, *Remote Tracking* provides for the tracking of workers on location from a central server. Remote tracking also allows workers in the field to view the location of all connected co-workers. Devices must be activated for tracking and have Internet connectivity through phone service for this feature to be enabled.

3 Gesture recognition project

3.1 Background

Human-computer interaction is moving away from the traditional mouse and keyboard. Motion-based gesture recognition is a versatile and intuitive form of interaction. For this project, students were challenged to develop a hand gesture recognition system using the acceleration sensing capabilities available on Sun Microsystems' Sun SPOTs shown in Figure 4. Further, the system was to be designed to allow other acceleration-sensing mobile devices, such as the Android, Windows Mobile HTC Fuse, or iPhone to be used as interaction devices to the system in the future.

Figure 4. Sun Microsystems' Sun SPOT.

3.2 Learning new theory and technologies

To prepare for this project the students had to become familiar with capabilities, limitations, and programming the Sun SPOTs shown above [10]. SPOTs (Small Programmable Object Technology) are small, wireless, battery-powered experimental sensor devices that can be easily adapted for wireless sensor network applications or novel user interfaces – the latter is used here. SPOTs have several on-board sensors including the 3D accelerometer that was used in this project to detect hand motion.

In addition to learning how to effectively program the SPOTs, students had to study several research papers dealing with acceleration-based gesture recognition. It was decided to take an approach used on the wiigee project [9] that utilized statistical Hidden Markov Models (HMM) to recognize gestures [7].

3.3 System functions

Requirements for the system included the ability to train gestures, recognize gestures, evaluate the effectiveness and accuracy of the gesture recognition, and an implementation of a simple game to demonstrate proof of concept.

The *Training* mode, shown in Figure 5, allows one user to connect a 3D acceleration sensing mobile device (e.g., Sun SPOT) and train the system with multiple gestures. The training data (x, y, z acceleration vectors) is captured, analyzed, and saved for later use in the recognition, evaluation, and demonstration modes.

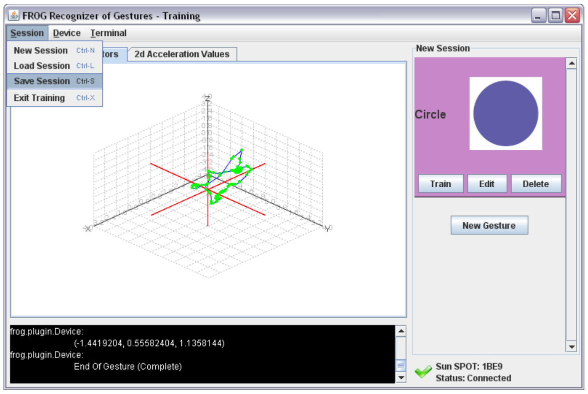


Figure 5. Training Mode.

The *Recognition* mode allows up to four users to simultaneously connect to the system and perform simple, previously trained gestures. Each user must load a library of trained gestures in order to perform recognition.

The *Evaluation* mode, shown in Figure 6, allows a user to connect and evaluate recognition accuracy. In this mode the system prompts the user to perform a specific gesture, accepts the gesture (x, y, z acceleration data) from the user, and then determines if the system is able to correctly identify the gesture performed. Statistics on number correct and incorrect are maintained allowing the user to determine the validity of his training set and reliability of the procedure.

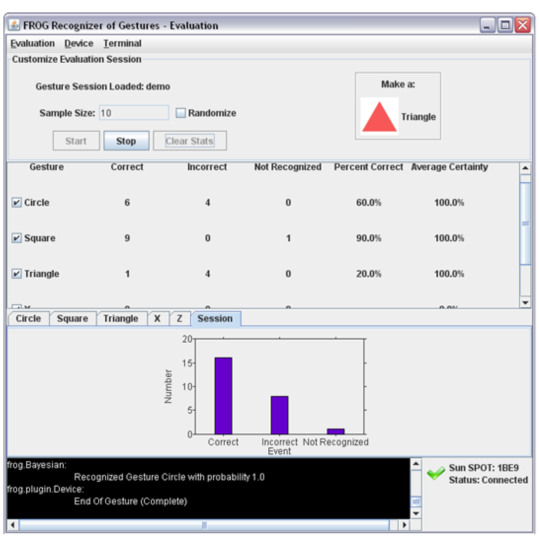


Figure 6. Evaluation Mode.

To provide a proof of concept, students were required to create a simple game that responded to hand gestures. The idea of the game was to prevent the UFOs from abducting the cows that were grazing in the pasture below. The UFOs were labeled with trained gestures (z, x, circle, square in Figure 7). The game player had to correctly perform the gesture matching the label before the UFO flew off-screen with the cow. Correct gesture recognition resulted in the UFO exploding, points for the player, and the cow being released. The game allowed up to four simultaneous players.

Figure 7. Demonstration Mode.

4 Results and observations

4.1 GIS project

The mobile GIS project was an extra ordinary project and student effort. The team members were very dedicated and committed to meeting deadlines and staying on schedule. Geographic information systems and map projections were new topics for the students and they had to take it upon themselves to research these subjects. In addition, mobile device programming with .NET and C# was a new programming environment for all team members and they embraced the opportunity to learn the technology. This project turned out to be a nice fit for the members of this team because it gave them the opportunity to apply concepts from upper level elective graphics and algorithms classes they had just completed.

As project sponsor, this author believes the student interest was significantly enhanced by affording the team the opportunity to program on mobile devices as they had no prior exposure to .NET or mobile device programming. In addition, this project had applicability outside the classroom and the students saw it as more than just a classroom exercise. They worked with real data obtained from a forestry management consultant. Just knowing that they were working on a real problem provided additional motivation.

This project was presented at the college's spring Student Research Symposium (SRS) and received a first place award for the department.

4.2 Gesture recognition project

This project served as the capstone senior project for the 2009-2010 academic year. As with the GIS project, it also represented an amazing effort on the part of team members and the project team leader. The work required to have all phases of the project operational was substantial and required excellent team effort and project management.

The makeup of the seven member team was such that there was a nice division of responsibilities. The project was particularly well suited for this senior class as there were two students with strong math backgrounds interested in investigating the training and recognition algorithms, other students were interested in sensors and SPOT communication issues, and still others interested in the interface design.

The students presented this project at an area student conference and at the college's Student Research Symposium (SRS) where it won a People's Choice Award and received a first place for the department. As with all senior capstone projects, the students also presented the project in a year-end capstone project presentation attended by their peers, faculty, family members, the department's Industry Advisory Board members, and other interested parties.

5 Conclusions

As previously discussed, there are several desirable requirements that should be considered for computer science capstone projects. With the exception of an external sponsor, the program at this university tries to address them all. And even this year, projects with external sponsors are being sought more aggressively.

In addition to the requirements traditionally emphasized, this author believes it to be very important that projects include a significant learning component – a new technology or new concepts. Students need to be able to learn on their own. The two projects included in this paper fulfilled that requirement.

Incorporating a significant learning component is essential, not only to provide additional motivation and student interest in projects, but perhaps more importantly, to instill in students a confidence in their ability to "dig out" information on their own. Graduates need to leave the program with confidence in their ability and knowledge that they are prepared to be lifelong learners.

6 References

[1] Buckley, M., et al. 2004. Benefits of Using Socially-Relevant Projects in Computer Science and Engineering Education. In *Proceedings of the SIGCSE '04* (Norfork, VA, Mar 3-7, 2004), 482-486. DOI= http://doi.acm.org/10.1145/971300.971463.

[2] Coppit, D., Haddox-Schatz, J. 2005. Large Team Projects in Software Engineering Courses. In *Proceedings of the SIGCSE '05* (St. Louis, MO, Feb 23-27, 2005), 137-141. DOI= http://doi.acm.org/10.1145/1047344.1047400.

[3] Dascalu, S., Varol, Y., Harris, F., and Westphal, B. 2005. Computer Science Capstone Course Senior Projects: From Project Idea to Prototype Implementation. In *Proceedings of the IEEE FIE-2005 Frontiers in Computer Education Conference* (Indianapolis, IN, Oct 19-22, 2005), S3J/1-6.

[4] ESRI. Environmental Systems Research Institute, Inc. http://www.esri.com.

[5] Gorka, S, Miller, J, and Howe, B. 2007. Developing Realistic Capstone Projects in Conjunction with Industry. In *Proceedings of the SIGITE '07* (Destin, FL, Oct 18-20, 2007), 27-32. DOI= http://doi.acm.org/10.1145/1324302.1324309.

[6] Hadfield, S. and Jensen, N. 2007. Crafting a Software Engineering Capstone Project Course. *Journal of Computing Sciences in Colleges*, 23(1), 190-197.

[7] Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77* (Feb 1989), 257-286.

[8] Roggio, R. F. 2007. Process-Driven Software Development: An Approach for the Capstone Sequence. In *Proceeding of ISECON 2007* (Pittsburg, PA, 2007), 1-11.

[9] Schlömer, T., Poppinga, B., Henze, N., and Boll S. 2008. Gesture Recognition with a Wii Controller. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction* (Bonn, Germany, Feb 18-20, 2008), 11–14. DOI=http://doi.acm.org/10.1145/1347390.1347395.

[10] Sun Microsystems Sun SPOT. http://www.sunspotworld.com.

Transforming Undergraduate Computer Science into Software Engineering

Kevin Daimi and Katy Snyder
Department of Mathematics, Computer Science and Software Engineering
University of Detroit Mercy
4001 McNichols Road, Detroit, MI 48221
{daimikj, snyderke}@udmercy.edu

ABSTRACT

The Software Engineering process involves a number of complex sub-processes including requirements engineering, analysis, design, testing, configuration, and maintenance. Designing a curriculum for an undergraduate Software Engineering program has never been an easy task. Converting an existing computer science program into a software engineering one is just as difficult. The goal of this paper is to introduce the process, justification, and details for transforming an existing undergraduate Computer Science program at the University of Detroit Mercy into a Software Engineering program.

Keywords

Computer Science, Education, Software Engineering, Transformation Process.

I. INTRODUCTION

Software Engineering is concerned with developing and maintaining reliable and quality-based software systems. These systems should be cost-effective to develop and maintain. It is essential that such systems satisfy all the customer's requirements to be qualified as useful and effective solutions for customers' problems [11], [15]. To achieve these goals, software engineering integrates the principles of mathematics, computer science, and management with engineering practices. Achieving such integration is a complicated process for curriculum designers. Curricula should help students develop skills from a variety of areas, including those listed above as well as software management, software design and development, communication, and critical thinking.

Because of the broad and interdisciplinary set of skills necessary to meet the needs of industry, the field of software engineering has become extremely complex [1]. When designing software engineering curriculum, the Software Engineering Body of Knowledge (SWEBOK) recommends ten knowledge areas to be included in the curriculum: software requirements, design, construction, testing, maintenance, configuration management, engineering management, engineering processes, engineering tools and methods, and quality [6]. Further recommendations for designing undergraduate software engineering curriculum are in the 2004 Software Engineering Curriculum Guide [14].

Bridging the gap between software engineering education and industry is an essential curriculum design criteria [8]. To accomplish that, collaboration between industry and academia throughout all the design stages is needed. In addition, tools deployed by industry should be extensively used within the program to prepare graduates, who are ready to use such tools. Software engineering practitioners have a great role to play in improving the future and the practical nature of software engineering education [9].

An industry prospective on software engineering education has been documented [13]. This author identified the need for software engineers who are problem solvers, possess excellent communication skills, and work collectively with users in an interdisciplinary environment. He added that universities should concentrate on providing students with knowledge and concepts, and leave the development of specific skills for the job field. In addition, universities should have more broadly defined learning objectives and establish a process to solicit industry inputs. While we agree with these viewpoints, we believe that some skills can still be acquired during students' time at the university if projects for various courses are provided by industry. This approach is more suitable for producing graduates who are ready to do the job with minimal additional guidance.

This paper describes a process for converting a current undergraduate degree program in Computer Science into Software Engineering. The Software Engineering program has the traditional math courses required for many engineering degrees in addition to a number of Electrical Engineering courses. The new Software Engineering program integrates software engineering with computer science, mathematics, hardware and networking, management, and entrepreneurship. It is important to enhance our software engineering programs with entrepreneurship as this supports the development of new computing markets, the introduction of new software and hardware technologies, and has the potential to create additional opportunities for employment [3]. In addition to software, requirements, testing, verification, security and integration, the transformed curriculum also includes courses on software quality, project management, professional world of work, engineering ethics, software ethics, technical writing, and software integration. In general, software engineering curriculum coverage is weak in the area of software integration [12]. This course was recommended by our Advisory Council, which consists of members from industry, health, government, and education.

II. TRANSFORMATION RATIONALE

During the academic year 2006-2007, the Bachelor of Science in Computer Science program went through a comprehensive program review. The Dean of the College of Engineering and Science (E&S), the external reviewer from academia, the external reviewer from industry, the University Undergraduate Review Committee, and the University Faculty Governance suggested that the department should develop a more focused curriculum for our undergraduate degree to increase enrollment despite the fact that computer science enrollments are down nationwide.

At the beginning of the academic year 07-08, three paths were explored to improve the program in order to make it more appealing and focused. The first path concentrated on developing interdisciplinary minors with other departments within and outside the E&S College. The first minor in Bioinformatics was created in collaboration with the Department of Biology. There are more plans in this regard.

The second path aimed at making freshmen programming classes more appealing and fun to students. A number of approaches, such as multimedia, game programming, and programming a robot are being considered to make programming more pleasurable. Currently, all of the projects in these freshmen classes are game-oriented. The intention is to aid in the recruiting process, to retain students, and to help provide motivation for students to put in the time and effort necessary to overcome the frustrations frequently faced by students as they are introduced to the discipline of developing quality code.

The third path, developing a more focused curriculum, was the most difficult task. The first suggestion was to replace the current program with a bachelor degree in Bioinformatics. This option was ruled out since it requires developing a number of new courses, and because of the inability to locate an accreditation body to accredit such a program. In addition, the department wanted to observe how the minor in Bioinformatics will proceed before switching to a degree in this field. Furthermore, it was felt that a graduate degree in Bioinformatics may be more appropriate than an undergraduate one.

Having considered the above options, the recommendation was to transform the computer science degree into software engineering. We looked at software engineering degrees at other universities, examined the Software Engineering Body of Knowledge (SWEBOK) introduced jointly by the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM), and analyzed the demand for Software Engineering. In addition, the committee consulted faculty at other universities, as well as our Department's Advisory Committee. It was determined that in order to focus on Software Engineering, only few courses needed to be developed in addition to courses in leadership and entrepreneurship. We were encouraged by the following findings:

- There is a high demand for software engineers in the market. According to the US Department of Labor, "Computer software engineers are one of the occupations projected to grow the fastest and add the most new jobs over the 2006-2016 decade," [10].
- For schools that offer degrees in Software Engineering, the minimum enrollment nationwide is 25 and the maximum enrollment is 343. Within the state of Michigan, the University of Michigan – Dearborn has an enrollment of 72 [5].
- Offering a degree in Software Engineering may allow the opportunity for students to receive a Professional Engineering (PE) license when available. Texas is the first State to grant PE in software engineering.

- There are 17 undergraduate programs in Software Engineering (SWE) that are accredited by ABET. In Michigan, University of Michigan – Dearborn is the only university that has its SWE program accredited [5]. It is anticipated that our program will acquire accreditation as well.
- The recommendation and support from our Advisory Council to pursue the transformation.
- The Institute of Electrical and Electronics Engineers (IEEE) offers the CSDP Certified Software Development Professional exam [2]. This is based on software engineering. The transformed program will prepare students for this exam.
- At our university, students look upon computer science as equivalent to programming. This fact has discouraged a number of students from enrolling in computer science. With software engineering, students will concentrate on developing the software product. To do that, they will be introduced to software requirements engineering, modeling and analysis, design, testing, and maintenance of software systems.
- There is some confusion regarding computer science and engineering among high school students, and to a certain extent, among admissions staff. Students in general, do not understand the difference between computer science and computer engineering. Both departments belong to the College of Engineering and Science, and a number of the same classes are required for both programs.
- Traditional undergraduate computer science programs' enrollments are decreasing nationwide. We believe that our focus on software engineering, in addition to making freshmen courses attractive and fun, will contribute to recruiting more students to the program.
- The focus on software engineering will further increase the collaboration with Electrical and Computer Engineering programs. Students will have more classes available from which to choose from. Furthermore, this collaboration will provide additional opportunities for students to do interdisciplinary projects. Electrical Engineering students can concentrate on the hardware part, and Software Engineering students will take care of the software part. The fact that students from both majors have been introduced to software and hardware will be quite valuable.
- The College of Engineering and Science offers a Doctor of Engineering (DEng.) degree. Currently, only students in Civil, Electrical, and Mechanical Engineering take advantage of it. Converting our programs to software engineering will make this degree available to our students as well.

III. THE FORMER PROGRAM

The Bachelor of Science in Computer Science was launched in 1976. This program was very successful in attracting students at the time it was introduced. However, it was mainly a liberal arts program rather than a professional program. To meet the requirements of industry, the program underwent a thorough revision in 1998. As a result of this revision, courses from Electrical Engineering, and Computer and Information Systems were added to the major requirements. In 2001, another review was carried out. Based on the feedback and comments from computer experts in the auto industry, recommended courses, such as Data Mining, Embedded Programming, and Distributed and Parallel Systems were added. In 2003, we added Software Engineering and Software Project Management based on the feedback we got from industry and other sectors that our graduates were lacking software engineering and leadership skills. With the growing need for security nationwide, courses in computer and network security were added in 2004. Finally, in 2006 Bioinformatics was added to serve the needs of researchers in Pharmaceutical Industry and Biomedical Engineering.

The former program was divided into four components; major requirements, technical electives, university core, and general electives. The Major Requirements included courses in computer science, mathematics, electrical and computer engineering, and technical writing with a total of 64 credits.

Students were required to take 12 credits of Technical Elective courses. These included courses in computer science, mathematics, and electrical and computer engineering. Students were also required to complete courses for the university core curriculum. These courses include: English and Fundamentals of Speech (ENL 1310, and CST 1010), two courses in natural science, PHL 1000 (Philosophy), one course in religious studies, any other course in philosophy or religious studies, three courses in history/literature/culture, one course in ethics, and one course in contemporary socio-political problems. These courses total at least 36 hours.

Finally, the remaining courses were general electives taken with the advisor's approval in the various areas where the student has special interests. This may include up to 9 credits of co-op. The student must complete 126 hours with at least a 2.0 QPR overall and a 2.0 QPR for the last 60 hours. A typical student will have 15 credits of general electives.

IV. THE TRANSFORMATION PROCESS

The process used to convert the BS in Computer Science program to a BS in Software Engineering has been comprised of the following steps:

1) Students were surveyed to get their feedback on switching to software engineering. The vast majority of students favored the idea of switching to software engineering. Students expressed their enjoyment of the few software engineering classes offered within the former program.
2) Faculty and administrators in the college were consulted as to the feasibility and desirability of developing the program.
3) The first draft of the proposed program was prepared.
4) The Department's Advisory Council was consulted to obtain their feedback.
5) The second draft was created.
6) Software engineering faculty at other universities were consulted to obtain their feedback on the second draft.
7) The third draft was created and verified against the SWEBOK [6] and the 2004 Software Engineering Curriculum Guide [14].
8) The findings were reflected in the fourth draft and discussed with the Advisory Council.
9) Details of the transformed program were outlined.
10) Met with the Department of Electrical and Computer Engineering (ECE) to discuss the collaboration regarding the ECE courses and the interdisciplinary Senior Design Project.
11) Rolled out the addition of the missing software engineering classes to the catalog.
12) Started offering the new courses. This effort started this academic year.
13) Completed the offering of all new classes.
14) Changed the name of the degree to Bachelor of Science in Software Engineering this winter.

V. THE TRANSFORMED PROGRAM

A. Program Highlights

1) This transformed program combines computer science, software engineering, computer engineering, management, entrepreneurship, problem solving, communication, and engineering and software ethics.
2) It provides the needed knowledge and foundational kills learned through industry-based projects.
3) It is a project-led program. Projects have a weight of at least 50%. Whenever possible, they are drawn from industry.
4) Software engineering courses are taught using the IBM Rational Software. The tools that have been adopted for coursework so far are; Rational Software Architect, Rational RequisitePro, and Rational Functional Tester. In addition to using the tools to implement students' projects, software tools enhance the teaching process and make courses more appealing to students.
5) Software engineering courses adopt the IEEE standards whenever possible.
6) The department has ongoing consultation with industry for continuous improvement.
7) The transformed program offers Software Integration, an important course demanded by industry.
8) The program is based on interdisciplinary projects whenever possible.
9) It follows the requirements of the Accreditation Board for Engineering and Technology (ABET).
10) Collaboration with Electrical and Computer Engineering, and Business is the cornerstone for the program.
11) Students in the program concentrate on only one programming language instead of three. If students are interested in learning more programming languages, they can take them as electives. Many computing professionals, who worked in industry years ago, were trained in one programming language. Once a new language was introduced and there was a need for it at the organization, it was well within workers' ability to learn the new syntax, since they already had mastered the underlying logic [4]. Furthermore, it has been concluded that prior programming experience is not necessary for students' success in software engineering [7].
12) The program allows for the development of Artificial Intelligence, Data Mining, and Bioinformatics software using the principles and techniques of software engineering. Software engineering practices are essential for creating

dependable software systems, especially in bioinformatics [16].

B. Program Objectives

The program is designed to provide educational excellence in software engineering and prepare graduates that are intellectually, spiritually, ethically, and socially developed to pursue a graduate degree in Software Engineering, or a career in industry, education, health, and government. The program objectives include:

1) Provide graduates with the level of technical skills needed for the professional practice of software engineering and for pursuing a graduate degree.
2) Prepare graduates who are intellectually, spiritually, and ethically developed to engage in lifelong learning.
3) Develop students' proficiency in oral and written communication, and teamwork effectiveness.
4) Instill within graduates the ability to demonstrate professional and ethical responsibility, entrepreneurship, leadership, and awareness of contemporary issues.

C. Program Outcomes

The program outcomes were designed to satisfy the requirements of ABET accreditation and the need of industry as recommended by our Advisory Council.

a) Students will be able to apply knowledge of mathematics, computer science, and software engineering to identify, analyze, model and solve software engineering problems, as well as measure the quality of software engineering solutions.
b) Students will be able to design and conduct software tests, as well as to analyze and interpret the test result data and outcomes.
c) Students will be able to develop, validate, and analyze software requirements for various software systems or components.
d) Students will be able to design, implement, and maintain a software system, component, or process to meet desired needs of stakeholders within a realistic set of quality constraints.
e) Students will have an ability to work in teams, including multidisciplinary teams, and lead software engineering projects.
f) Students will have an ability to identify, formulate, analyze, and solve software engineering problems.
g) Students will develop an understanding of professional standards and ethical responsibility needed for practicing software engineering.
h) Students will be able to effectively communicate their software engineering outcomes both orally and in writing.
i) Students will be able to gain the broad education necessary to understand the impact of software engineering solutions in a global, economic, environmental, and societal context.
j) Students will recognize the need for, and an ability to engage in life-long learning.
k) Students will be able to acquire the knowledge of contemporary issues and understand the impact of software engineering on these issues.
l) Students will able to use the techniques, knowledge, and modern software engineering tools necessary for software engineering practice.
m) Students will be able to apply entrepreneurship skills to software engineering and develop leadership skills.

D. Program Requirements

The program requirements are summarized as follows:

- Major Requirements 76 cr.
- Technical Electives 06 cr.
- University Core 36 cr.
- General Electives 08-09 cr.

The Major Requirements include a common Computing and Software Engineering courses core. Students will also take 6 hours of Technical Electives. CSSE-4610 Data Mining and CSSE-4620 Bioinformatics must be taken if the Bioinformatics Minor is pursued. Courses for the common Computing/Software Engineering core, Math and Computer Engineering requirements, and the Technical Electives are provided in Tables 1-3.

The University Core is the same as in the former program with the exception of replacing the traditional ethics classes with three courses in Professional World of Work (ENGR 3010, ENGR 3020, and ENGR 3030), ENGR-1000 Ethics and Politics of Engineering, and CSSE-2750 Software Ethics. For the General Electives, students can take any 8-9 credits as General Electives.

TABLE I
COMPUTING & SOFTWARE ENGINEERING COURSES

Course	Credits
CSSE-1710 Computer Science I	3
CSSE-1720 Computer Science II	3
CSSE-3150 Software Engineering	3
CSSE-3540 Database Systems	3
CSSE-4160 Software Testing	3
CSSE-4390 Software Quality Engineering	3
CSSE-4400 Software Requirements Engineering	3
CSSE-4430 Data Structures	3
CSSE-4490 Operating Systems	3
CSSE-4540 Computer and Software Security	3
CSSE-4560 Software Systems Verification	3
CSSE-4570 Software Project Management	3
CSSE-4860 Software Integration	3
CSSE-4910 Software Product Entrepreneurship	3
CSSE-4930 Senior Design Project	3
ENL-3030 Technical Writing	3

TABLE II
MATH AND COMPUTER ENGINEERING COURSES

Course	Credits
MTH-1410 Calculus I	4
MTH-1420 Calculus II	4
MTH-2760 Discrete Mathematics	3
MTH-4020 Linear Algebra	3
MTH-4270 Applied Probability and Statistics	3
ELEE-3640 Digital Logic Circuits	3
ELEE-3650 Digital Logic Lab.	1
ELEE-4680 Computer Networks	3
ELEE-4690 Computer Networks Lab	1
ELEE-4800 Computer Architecture	3

TABLE III
TECHNICAL ELECTIVES

Course	Credits
CSSE-3270 User Interface Engineering	3
CSSE-3470 Software Configuration	3
CSSE-3480 Software Maintenance	3
CSSE-4411 Perl	3
CSSE-4130 Java	3
CSSE-4530 Web Software Engineering	3
CSSE-4531 Software Architecture	3
CSSE-4535 Systems Engineering	3
CSSE-4550 Artificial Intelligence	3
CSSE 4610 Data Mining	3
CSSE-4620 Bioinformatics	3
CSSE-4640 Parallel Systems	3
CSSE-4670 Distributed Systems	3
CSSE-4900 Special Topics in SWE	3
MTH-4510 Advanced Calculus	3
ELEE-3860 Introduction to Microcontrollers	3
ELEE-3870 Microcontrollers Lab	1
ELEE-4640 Hardware Description Languages	3
BUS-3180 Organizational Leadership	3

E. Program Assessment

Program assessment is an essential component when designing a curriculum. It is an iterative and incremental process that plays a key role in measuring learning and improving programs [17]. The activities that will be conducted for this purpose include:

1) Individual course assessment will be completed to ensure that each course is achieving its learning outcomes and supporting the program outcomes.
2) A comprehensive program self study will be prepared for the purposes of ABET accreditation.
3) An Undergraduate Survey will be employed to measure students' satisfaction with individual courses and the program as a whole.
4) A survey for the students taking the capstone design course (Senior Design Project) will be administered. It will assess how well students believe the program is achieving its learning outcomes and how well students' learning experience matches the program objectives.
5) A Comprehensive Test will be devised to measure how well students have achieved the learning outcomes. This test will be offered as part of the capstone course during the last term of study with a weight of 30%. It will involve the following courses, Software Engineering, Software Testing, Software Quality Engineering, Software Requirements Engineering, Software Verification, Software Integration, and Computer and Software Security.
6) An Exit Survey will be offered to students completing the program to solicit their feedback on the program and on how to improve it.
7) An Alumni Survey will be used to discover how well our graduates feel they were prepared for their current position.
8) An Employer Survey will be prepared to obtain the feedback of employers on how well our graduates are prepared for their positions.

VI. CONCLUSION

Creating a new program from scratch is an extensive process, and often the final approval is neither straightforward nor guaranteed. Approval must be secured from a number of committees at the department, college and university levels, in addition to a number of administrators. Finally, the proposal must be approved by the university board. Instead of following this traditional approach, a transformation process to convert the current computer science program into software engineering was followed. This approach frequently requires more planning than developing a brand new program; however, the transformation process can be completed in a shorter time frame, and is more likely to gain approval. The proposed program has been approved and the department is allowing current students to switch to the new program with the exception of seniors graduating this year.

REFERENCES

[1] M. Bernhart, T. Grechenig, J. Hetzl, and W. Zuser, “Dimensions of Software Engineering Course Design,” in *Proc. 28th International Conference on Software Engineering (ICSE’06)*, Shanghai, China, 2006, pp.667–672.

[2] CSDP Certified Software Development Professional, IEEE Computer Society, Available: http://www.computer.org/portal/web/certification.

[3] K. Daimi, and N. Rayess, “The Role of Software Entrepreneurship in Computer Science Curriculum,” in Proc. *International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS’08)*, Las Vegas, Nevada, 2008, pp 53-62.

[4] K. Daimi, “A Framework for Revitalizing Undergraduate Computer Science,” in Proc. *International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS’08)*, Las Vegas, Nevada, 2008, pp 332-337.

[5] Engineering and Technology Enrollments, Engineering Workforce Commission of the American Association of Engineering Societies, Fall 07, Available: http://www.ewc-online.org.

[6] *Guide to Software Engineering Body of Knowledge*,” IEEE, 2005, Available: http://www.swebok.org.

[7] M. Hamilton, and L. Haywood, “Learning About Software Development – Should Programming always Come First?” in *Proc. 6th Australasian Computing Education Conference*, Dunedin, 2004, pp.131–136.

[8] S. Honiden, Y. Tahara, N. Yoshioka, K. Taguchi, and H. Washizaki, “Top SE: Educating Superarchitects Who Can Apply Software Engineering Tools to Practical Development in Japan,” in *Proc. 29th International Conference on Software Engineering (ICSE’07)*, Minneapolis, 2007, pp.708–718.

[9] T .C. Lethbridge, J. Diaz-Herrera, R. J. LeBlanc Jr., and J. B. Thompson, “Improving Software Practice Through Education: Challenges and Future Trends,” in *Proc. Future of Software Engineering (FOSE’07)*, Minneapolis, 2007, pp.12–28.

[10] Occupational Outlook Handbook, 2008-09 Edition, Available: www.bls.gov/oco/ocos267.htm.

[11] S. L. Pfleeger, and J. M. Atlee, *Software EngineeringTheory and Practic*. New York, NY: Prentice Hall, 2010, pp. 1–44.

[12] H. Reichgelt, and V. Jovaovic, “Software Management as an Information Technology Knowledge Area,” in *Proc. 4th Conference on* Information Technology Curriculum *(CITC4’03)*, Lafayette, Indiana, 2003, pp.31–36.

[13] D. J. Reifer, “Educating Software Engineers: An Industry Viewpoint,” *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 3, pp. 8–9, May 2005.

[14] Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, IEEE/ACM Joint Task Force on Computing Curricula, 2004, Available: http://sites.computer.org/ccse.

[15] I. Sommerville, *Software Engineering*. New York, NY: Addison Wesley, 2006, pp. 1–19.

[16] M. Umarji, M. Pohl, C. Seaman, A. G. Koru, and H. Liu, “Teaching Software Engineering to End-Users,” in *Proc. 4th Workshop on End User Software Engineering (WEUSE IV’08)*, Leipzig, Germany, 2008, pp.40–42.

[17] T. Wang, D. Schwartz, and R. Lingard, “Assessing Students Learning in Software Engineering,” *Journal of Computer Sciences in Colleges*, vol. 23, no. 6, pp. 239–248, June 2005.

LED: A formal language for defining computable functions

J. Nelson Rushton [1], **Jarred Blount**[1]
[1]Computer Science, Texas Tech University, Lubbock TX USA

Abstract **-** Syntax and semantics are given for the the Language of Effective Definitions (LED), a formal language for defining computable functions. LED is a declarative language in a hybrid functional/logical paradigm. Its scalar data are symbols, (exact) rational numbers, and lambda expressions; and its nonscalars are finite sets and tuples. LED is suitable for use as a specification language, programming language, or for teaching concepts of mathematical definition – such as recursion, piecewise definition, quantification, and set comprehension – in a lightweight, but rigorously specified framework. Its formal syntax is lisp-like, but it is intended to be used as a meta-language for more human-friendly representations.

Keywords: *A Maximum of 6 Keywords*

1 Introduction

This paper defines the Language of Effective Definitions (LED), a language for defining computable functions. The goal is to provide a medium of communication for effective mathematical definitions which is (1) rigorously specified, (2) lightweight, i.e., definable in a few pages, and (3) uses the same data types and primitive constructs as informal mathematics. LED's formal syntax is lisp-like, but it can be used as a meta-language for more human-friendly representations, which are customizable as the situation requires (perhaps including Unicode or even LaTeX source code).

The paper is organized as follows: Section 2 defines syntax; Section 3 gives semantics. Section 4 is an example program, implementing an AI strategy for tic-tac-toe.
Instructions for authors

2 Formal Syntax

All LED tokens are strings of printable, non-white-space ascii characters. The token set is defined as follows:

- A *symbol* is a string of non-white-space characters not beginning with a digit and containing no left or right parentheses or quotation marks (").
- A *reserved symbol* is one of the following strings: `+ - / \ * ^ @ = < > <= >= + nil and or not lambda set tuple some all setof union intersect intdiv def if owise`
- A *digit string* is a string of one or more digits.
- A *nonnegative numeral* is either a digit string, a period (`.`) followed by a digit string, or a digit string followed by a period and then another digit string.
- A *numeral* is either a nonnegative numeral or a nonnegative numeral preceded by a minus sign (`-`).
- A *quoted string* is an arbitrary sequence of printable characters enclosed in quotation marks (").
- A *token* is either a symbol, a reserved symbol, a numeral, or a quoted string.tt

An *expression* is either a token, or a sequence of one or more expressions, separated by white space and enclosed in parentheses. A *definition* is an expression of one of the following forms:

- (`def` (f $x_1…x_n$) e) where $n \geq 0$, f and the x_i are symbols, and e is an expression,
- (`def` (f $x_1…x_m$) (`if` g_1 h_2)…(`if` g_n h_n)),) where $n \geq 0$, f and the x_i are symbols, and the g_i and h_i are expressions; or
- (`def` (f $x_1…x_m$) (`if` g_1 h_2)…(`if` g_n h_n) (`otherwise` h_{n+1})) where $n \geq 0$, f and the x_i are symbols, and the g_i and h_i are expressions.

In each of case, f is the *defined symbol* of the definition and x_i are its *parameters.*. A *program* is a set of definitions, and a definition which is an element of program P is called a *definition of P*. Defined symbols of definitions of P may be called *function symbols of P*. An occurrence in P of a symbol which is not a function symbol of P is an occurrence *as a variable.* Parameters of definitions must occur as variables – that is, no function symbol of P can be used as a parameter in a definition of P.

If $x_1,…,x_n$ are symbols, a *literal schema in* $x_1,…,x_n$ is either a numeral, a symbol, a lambda expression, or (`tuple` e_1 ... e_n) where $n \geq 2$ and each e_i is a literal schema in $x_1,…,x_n$.

All variable occurrences in a program must be bound. Variables are *bound* as follows:

- The parameters of a definition are bound in within that definition.
- The symbols $x_1...x_n$ are bound in (`lambda` ($x_1...x_n$) e).
- All symbols which occur as variables in expression e are bound in (`in` e S) .
- If e is a literal schema in $x_1,…,x_n$, all occurrences of the x_i's are bound in (= e_1 e).
- If x is bound in p, it is bound in (and p q), (`some` p q), (`all` p q), and (`setof` q p).

- If x is bound in h, then it is bound in (`if` g h).

A *substitution* is a finite set of pairs (x, e) where x is a symbol and e is an expression, such that no variable occurs as the first coordinate of more than one such pair. If e is an expression and s is a substitution, then e/s denotes the expression obtained from e by replacing all free occurrences of x with g whenever $(x,g) \in e$. The substitution of e for x in p is *safe* if no free occurrence of a variable in e becomes bound when it is substituted for x in p. All substitutions described in the semantics are presumed to be safe, by renaming variables whenever necessary.

For n>2 the expressions (`and` $p_1 \ldots p_n$) and (`or` $p_1 \ldots p_n$) may be written as shorthands respectively, for (`and` …(`and` (`and` p_1 p_2) p_3) …p_n) and (`or` …(`or` (`or` p_1 p_2) p_3) …p_n). The quoted string "$a_1a_2 \ldots a_n$" is shorthand for (`set` (`tuple` 1 c_1)…(`tuple` n c_n)), where ai is the ascii code of ci.

2.1 Length

The maximum allowed number of pages is seven for Regular Research Papers (RRP) and Regular Research Reports (RRR); four for Short Research Papers (SRP); and two for Posters (PST).

3 Semantics

A *datum* (pl. *data*) is either a rational number, an atom, a lambda expression, a finite set of data, or an n-tuple of data where $n \geq 2$. The values of well defined LED expressions, when interpreted under a given program, are data.

We will say that *e denotes x under P* if, roughly speaking, expression e denotes a datum x in the context of the definitions in program P. This is defined precisely by (the minimal Herbrand model of) the following mutually recursive semantic rules. If e denotes x under P, and the program P is clear from context, then we may simply say that *e denotes x*.

For the remainder of this section let P be a fixed program. The meta-variables e and g will vary over expression; x will vary over symbols; a, b, c, n, and f will vary over data; and s will vary over substitutions.

General definitions:

1. *defined:* We will say that expression e is *defined* if there exists a datum r such that e denotes r.
2. *true and false:* We will say that *e is true* if e denotes the symbol `true`. Similarly we will say that *e is false* if e denotes the symbol `false`.
3. *propositions:* A *proposition* is an expression which is either true or false.
4. *solution*: If $x_1,\ldots,x_n$ are symbols and e is an expression, a *solution of e for* $x_1,\ldots,x_n$ is a substitution s, whose domain is the set of symbols $\{x_1,\ldots,x_n\}$, such that e/s is true. When a set of symbols$\{x_1,\ldots,x_n\}$ is clear from context, then a solution in $x_1,\ldots,x_n$ of e may simply be called a *solution of e*.

Scalars:

5. *rational numbers:* If e is a numeral which denotes the rational number a under the usual reading in base 10 notation, then e denotes r.
6. *symbols and lambda expressions:* If e is a symbol or lambda expression, then e denotes itself.

Rational arithmetic:

7. *basic arithmetic operations:* If e_1 and e_2 denote rational numbers a and b, then (+ e_1 e_2) denotes the rational number a+b. Similarly for (- e_1 e_2), (- e_1), and (`*` e_1 e_2). If b is not 0, then similarly for (/ e_1 e_2). In case a and b are not both 0, then (`^` e_1 e_2) denotes the rational number a^b.
8. *integer division and modulus:* If e_1 denotes the nonnegative integer n_1, e_2 denotes positive integer n_2, and k is the greatest integer such that $kn_2 \leq n_1$, then (`intDiv` e_1 e_2) denotes k and (`mod` e_1 e_2) denotes n_1-kn_2.
9. *floor:* If e denotes a rational number a, then (`floor` e) denotes the greatest integer which does not exceed a.
10. *numeric comparisons:* If e_1 and e_2 denote rational numbers a and b respectively, then (`<` e_1 e_2) is true if $a < b$ and false otherwise. Similarly for (`>` e_1 e_2), (`<=` e_1 e_2) and (`>=` e_1 e_2).

Data structures:

11. *sets:* If $n \geq 0$ and e_i denotes a_i for $1 \leq i \leq n$, then (`set` e_1 ... e_n) denotes the set $\{a_1,...,a_n\}$.
12. *tuples:* If $n \geq 2$ and e_i denotes a_i for $1 \leq i \leq n$, then (`tuple` e_1 ... e_n) denotes the tuple $(a_1,...,a_n)$.

Equality:

13. If e_1 and e_2 are defined then (= e_1 e_2) is true if e_1 and e_2 have the same denotation and false otherwise.

Set operations:

14. *membership:* If e_1 denotes a datum a and e_2 denotes a set b, then (`in` e_1 e_2) is true if $a \in b$ and false otherwise.
15. *basic set operations:* If e_1 and e_2 denote sets a and b respectively, then (`union` e_1 e_2) denotes the set $a \cup$

b, (`intersect` e_1 e_2) denotes $a \cap b$, and (`\` e_1 e_2) denotes the set difference $a \backslash b$.

Boolean connectives:

16. *and:* If e_1 is false then (`and` e_1 e_2) is false. If e_1 is true and e_2 is a proposition then (`and` e_1 e_2) denotes the value of e_2.
17. *or:* If e_1 and e_2 are propositions then (`or` e_1 e_2) is false if both e_1 and e_2 are false, and true otherwise.
18. *not:* If e is a proposition then (`not` e) is true if e is false, and false if e is true.

Simply solvable expressions:

19. *propositions:* If e is a proposition and $x_1,\ldots,x_n$ are symbols, then e is simply solvable for $x_1,\ldots,x_n$
20. *equations:* If e is a literal schema in $x_1,\ldots,x_n$ and e_1 is defined, then (= e_1 e) is simply solvable for $x_1,\ldots,x_n$.
21. *set membership:* If e_1 is a literal schema in $x_1,\ldots,x_n$ and e_2 denotes a set, then (`in` e_1 e_2) is simply solvable for $x_1,\ldots,x_n$.
22. *conjunction:* If e_1 is simply solvable for $x_1,\ldots,x_n$ and for every solution s of e_1 we have that e_2/s is simply solvable for $x_1,\ldots, x_n$, then (`and` e_1 e_2) is simply solvable for $x_1,\ldots,x_n$.
23. *disjunction:* If e_1 and e_2 are both simply solvable for $x_1,\ldots,x_n$ then (`or` e_1 e_2) is simply solvable for $x_1,\ldots,x_n$.

Quantification and set comprehension:

24. *all:* Suppose x is a symbol, e denotes a set, and for every solution s of (`in` x e) for x we have that p/s is a proposition. Then (`all` (`in` x e) p) is true if p/s is true for every solution s of (`in` x e) for x, and false otherwise.
25. *some:* Suppose x is a symbol, e denotes a set, and for every solution s of (`in` x e) for x we have that p/s is a proposition. Then (`some` (`in` x e) p) is true if p/s is true for some solution s of (`in` x e) for x, and false otherwise.
26. *set comprehension:* If e_2 is simply solvable for $x_1\ldots x_m$, no x_i is a function symbol, and e_1/s is defined for every solution s of e_2 for $x_1\ldots x_m$, then (`setof` e_1 e_2) denotes the set $\{a \mid s$ is a solution of e_2 and e_1/s denotes $a\}$.

Lambda expressions:

27. *beta reduction:* If $e_1,\ldots,e_m$ are well defined and $e/\{(x_1,e_1),...,(x_m,e_m)\}$ denotes a, then ((`lambda` $(x_1...x_m)$ e) $(e_1 \ldots e_m)$) denotes a.

Definitions:

28. *unconditional:* If (`def` (f $x_1\ldots x_m$) e) is definition of P, where e is an expression, $e_1\ldots e_n$ are defined, and $e/\{(x_1,e_1),..., (x_k.e_m)\}$ denotes a, then (f $e_1\ldots e_n$) denotes a.
29. *conditional:* If (`def` (f $x_1\ldots x_m$) (`if` g_1 h_2)…(`if` g_n h_n)) is a definition of P, $e_1,\ldots,e_m$ are all defined, $s=\{(x_1,e_1),\ldots,(x_m,e_m)\}$, $g_1/s,\ldots g_n/s$ are all simply solvable, j is the only element of $\{1,\ldots,n\}$ such that g_j/s has a solution, and $h_j/s/s'$ denotes a for every solution s' of g_j/s, then (f $e_1\ldots e_m$) denotes a.
30. *otherwise* If (`def` (f $x_1\ldots x_m$) (`if` g_1 h_2)…(`if` g_n h_n) (`otherwise` h_{n+1})) is a definition of P, $e_1,\ldots,e_m$ are all defined, $s=\{(x_1,e_1),\ldots,(x_m,e_m)\}$, $g_1/s,\ldots g_n/s$ are all simply solvable, there is no element j of $\{1,\ldots,n\}$ such that g_j/s has a solution, and h_{n+1}/s denotes a, then (f $e_1\ldots e_m$) denotes a.

4 Sample Code

This section gives sample LED describing the rules of tic-tac-toe, and a simple AI strategy for the game.

We begin by defining the cross product of sets, using set comprehension:

```
(def (cross S T)
  (setof (tuple x y)
     (and (in x S) (in y T))))
```

The definition of set cardinality demonstrate the recursive traversal of a set. The traversal may happen in any order, and so well definedness needs to be proven in case it is not obvious (in this case it is obvious). Note that `(set)` denotes the empty set.

```
(def (cardinal S)
  (if (= S (set)) 0)
  (if (in x S)
      (+ 1 (cardinal (\ S (set x))))))
```

Next we define the game board, and the rows which form a winning “three-in-a row”.

```
(def game_board
  (cross (set 1 2 3) (set 1 2 3))


(def hrows
 (setof
  (cross (set i)  (set 1 2 3))
  (in i (set 1 2 3)))))
```

```
(def vrows
  (setof (cross (set 1 2 3)
      (set i)) (in i (set 1 2 3)))))

(def diagonals
  (set
    (set (tuple 1 1)
    (tuple 2 2)
    (tuple 3 3))
  (set
    (tuple 3 1)
    (tuple 2 2)
    (tuple 1 3))))

(def rows
  (union (union hrows vrows) diagonals))
```

A player wins by getting three in a row. If the board fills up without either player winning, then the game goes to the proverbial “cat”. In either case, the game is over.

```
(def (three_in_row G p)
  (some
    (in r rows)
    (all (in c r)
    (in (tuple p c) G))))

(def (cat G)
  (and (and
    (=(cardinal G) 9)
    (not (three_in_row G “x”)))
    (not (three_in_row G “o”))))

(def (over G)
  (or
   (three_in_row G “x”)
   (three_in_row G “o”)
   (cat G)))
```

If the game is not over, then it is either *x*’s turn or *o*’s turn, as determined by the parity mod 2 of the number of moves already made:

```
(def (turn G p)
  (and
    (not (over G))
    (or
      (and
        (= p “x”)
        (= (mod (cardinal G) 2) 0))
      (and
        (= p “o”)
        (= (mod (cardinal G) 2) 1)))))
```

It is legal in game state *G* for player *p* to move in cell *c* if it is *p*’s turn in *G* and cell *c* is empty in *G*.

```
(def (legal_move G p c)
 (and
  (turn G p)
  (all
    (in (tuple c2 p2) G)
    (not (= c2 c)))))
```

Now for the AI strategy. The *successors* of *G* are the game states which may come immediately after G in a legal game.

```
(def (successors G)
  (setof
    (union G (tuple c p))
    (and
      (in c game_board)
      (in p (set “x” “o”))
      (legal_move G p c))))
```

The opponent of *x* is *o*, and vice versa.

```
(def (opponent p)
  (if (= p “x”) “o”)
  (if (= p “o”) “x”))
```

Finally, we define the AI predicate *S*. It can be shown that (`S G p`) holds if and only if player *p* has a strategy for achieving a win or tie from state *G*. The proof is by induction on the number of empty cells in *G*.

```
(def (S G p)
  (or
    (three_in_row G p)
    (cat G)
    (and
      (turn G p)
      (some
        (in G’  (successors G))
        (S G’ p))))
     (and
        (turn G (opponent p))
        (all
          (in G’ (successors G))
          (S G’ p)))))
```

OSSIE: An Open Source Software Defined Radio Platform for Education and Research

J. Snyder[1], **B. McNair**[2], **S. Edwards**[1], **and C. Dietrich**[2]
[1]Department of Computer Science, Virginia Tech, Blacksburg, VA
[2]Bradley Department of Electrical Engineering, Virginia Tech, Blacksburg, VA

Abstract— *OSSIE (Open Source SCA Implementation::Embedded) was originally released in 2004 to fill the need for an implementation of the Software Communications Architecture (SCA) [4] that was free, simple, open to modifications, and written in C++. OSSIE includes several features, including the core framework for running SDR applications called waveforms, a library of pre-made waveforms and components for creating new waveforms, the Waveform Workshop, a set of tools for rapid prototyping and debugging of waveforms, and a set of lab assignments designed around OSSIE. With these features OSSIE is suitable for both classroom instructional usage and as a platform for a wide range of research, from things as small as a one-semester undergraduate independent study to a PhD dissertation. This paper describes a project done for a Summer independent study, and shows how OSSIE was used to create, run, and test the software for the project.*

Keywords: Software Defined Radio, Undergraduate Research Experiences

1. Introduction

The SCA is is an architectural framework with a set of design constraints that was created to maximize portability, interoperability, and reconfigurability of software radios while still allowing enough flexibility to address domain specific requirements and restrictions. Under the SCA, software radios run programs called waveforms that are composed of individual signal processing modules called components.

OSSIE [3], [14], [9] was created to serve as an open source implementation of the SCA to enable SDR and cognitive radio education and research. It has also been used for rapid prototyping and proof-of-concept and reference implementations. These include SDR waveforms for public safety and vehicular applications [12], [11], as well as SDR interfaces [6], abstraction approaches [10] and architectures [13]. To that end, OSSIE consists of four main parts:

- The Core Framework
- The Waveform Workshop
- A library of pre-made component and waveforms
- A set of lab assignments designed to work with OSSIE

1.1 The Core Framework

The core framework consists of all of the underlying code responsible for running waveform applications. In OSSIE, the core framework uses a middleware called CORBA [1] to facilitate communication between waveforms and components. Along with running waveforms, the framework itself makes a good subject for researching communications architectures [5], [18], [16].

1.2 The Waveform Workshop

The SCA defines the form of components and waveforms, including both necessary features in component source code and XML descriptors for components and waveforms. While this information is important to ensure that components are able to communicate and waveforms are able to run correctly, it has little to do with the actual functionality of the components or waveforms. In order to allow developers to quickly create new components and waveforms without having to know all of the details required by the SCA, the OSSIE Waveform Developer (OWD) was created [8]. When using OWD, the developer only had to write the code for the desired functionality of the component. OWD took care of generating all of the necessary infrastructure details. In recent years, OWD has been built into the OSSIE Eclipse Feature (OEF) [17], a plugin for the Eclipse IDE, adding features such as built-in support for version control and debugging.

Along with OEF, The Waveform Workshop includes two other tools: ALF and The Waveform Dashboard (WaveDash) [15]. ALF is a visual debugging tool that displays running waveforms as block diagrams, allowing the user to inspect data as it is passed from component to component by plotting the data or, if the data is audio, playing it through a speaker. The Waveform Dashboard is a GUI tool that allows the user to interact with running waveforms by changing the values of component properties in real time. WaveDash abstracts the SCA interfaces and implementation of the OSSIE core framework and provides a direct manipulation interface to interact with SDR waveforms. In addition, WaveDash was purposely written with a separable controller and GUI. This allows the controller to be used as a part of other applications.

1.3 A Library of Components and Waveforms

OSSIE includes a library of pre-made components and waveforms that can be used to create new waveforms and SDRs. The library includes many signal processing components, such as amplifiers, simulated channels, and demodulators. Waveforms include a simple demo, an AM receiver, and a digital voice transceiver. Many of the components and waveforms work with a Universal Software Radio Peripheral (USRP) [2] to transmit and receive signals as well as performing analog-to-digital and digital-to-analog conversion.

1.4 Lab Assignments

Finally, OSSIE includes a set of lab assignments designed to demonstrate OSSIE's features and illustrate how to use them. The series of labs starts with recreating, installing, and running one of OSSIE's simplest included waveforms. Subsequent labs include creating new components and waveforms, using OSSIE with a USRP, and finally creating a distributed waveform that runs on multiple networked computers. The labs walk the users through using OSSIE step by step, engaging them in hands-on learning in preparation for more extensive projects.

1.5 OSSIE's Role in Education

Between the core framework, the Waveform Workshop, and the included components, waveforms, and labs, OSSIE makes an ideal platform for SDR education and research. In the classroom, the labs provide a solid introduction to SCA and SDR concepts and prepare students for further SDR study. For research at the undergraduate level, the Waveform Workshop abstracts enough of the details that the learning curve associated with learning about SDR is greatly diminished. Students can get up to speed quickly, and can start making custom waveforms from pre-made components almost immediately. With knowledge of the Python programming language, students can create custom SDR applications using the WaveDash controller. At the graduate level, students can go more in-depth, working with the core framework itself, or simply using OSSIE as a platform, creating components and waveforms to test new signal processing techniques.

This paper describes a Summer independent study completed by an undergraduate student at Virginia Tech. The goal of the project was to design, develop, and demonstrate a proof-of-concept SCA-based multimode SDR transceiver than can quickly be reconfigured to operate using multiple modulation techniques. To do this, the student used OSSIE to create a software application that could launch and control an AM receiver and an FM transceiver.

2. Creating the Waveforms

2.1 Design

The student designed and implemented one waveform and implemented a second waveform for the project. The first waveform was an AM receiver. As seen in figure 1, the waveform consists of 6 components deployed to a general purpose processor (GPP). The USRP_Commander sets the receive frequency, decimation rate, and other parameters of the USRP. The decimator receives the signal from the USRP at 250 ksps and reduces the sampling rate by a factor of 10 to 25 ksps. The signal then continues to the automatic gain control (AGC). The AM demodulator receives the signal and outputs the demodulated signal. An amplifier component provides audio gain and sends the amplified signal to the sound card.

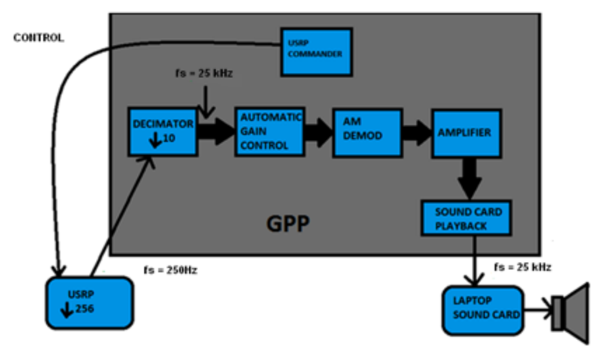


Fig. 1: AM Receiver Design

The second waveform was an FM transceiver that was implemented based on a waveform and components that a graduate student had implemented for an SDR class using an earlier version of OSSIE. As shown in figure 2, the FM waveform consists of 10 components deployed to a GPP. Transmit and receive are the two different paths associated with the FM waveform. When path select is set to one, it behaves as a transmitter; on the other hand, when the path select is changed to zero, it behaves as a receiver. In receiver mode, the signal reaches the decimator and reduces the sampling rate by a factor of 10 to 25 ksps. A WFM demodulator receives the signal and outputs the demodulated signal to the amplifier. Finally, the signal reaches the soundcard and the transmission is completed. Likewise, when the transceiver behaves like a transmitter the signal is first amplified. Then, in order to increase the effective sampling rate the signal is then interpolated. Finally, the Frequency Modulator receives the signal and outputs the demodulated signal to the USRP for transmission.

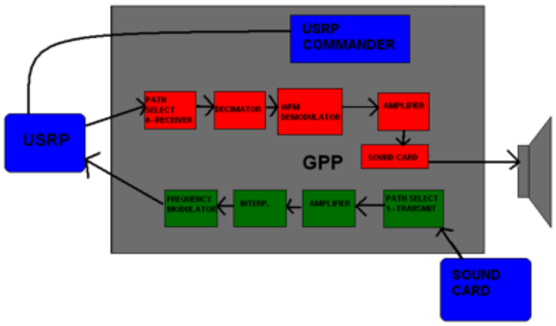


Fig. 2: FM Transceiver Design

2.2 Implementation

Both of these waveforms were implemented using OEF. Figure 3 shows the main OEF interface when creating waveforms. To implement a waveform using existing signal

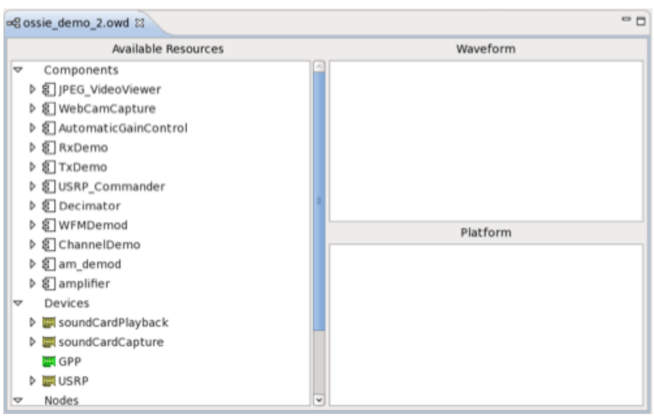


Fig. 3: OEF Interface for creating Waveforms

processing components and devices, the developer first drags all of the needed components from the Available Resources window to the Waveform window. Once there, the components can be expanded to display their ports. Input and output ports can then be dragged together to make connections (see figure 4). Once all of the necessary port connections are made, the waveform is essentially finished and only needs to be built and installed. With the original OWD, this step had to be performed manually using the automake tools. One of the major improvements of the switch to OEF was automatic building and installation. Every time the waveform is saved with OEF, it is rebuilt and reinstalled.

3. Creating the Application

After creating the two waveforms, the next step in the project was to create an application with a GUI to control them. This process was broken down into three phases.

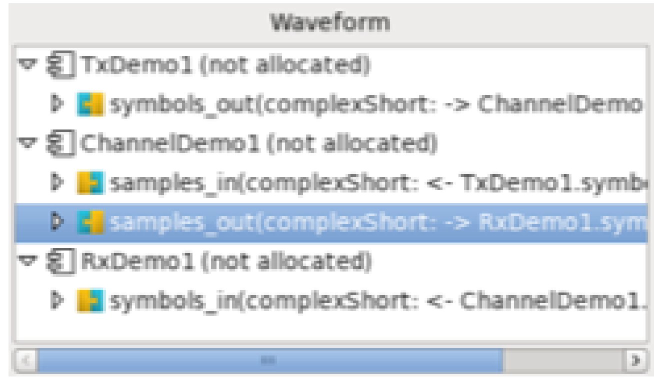


Fig. 4: Connecting Ports in OEF

3.1 Phase I - Learning the WaveDash Controller

The first phase was spent getting familiar with the WaveDash controller. The controller is the part of Wavedash that handles all of the communication with running waveforms. By leveraging the controller, you can programmatically install and control waveforms from Python applications. For this project, the student studied WaveDash itself and other applications written by the OSSIE team that make use of the WaveDash Controller, and then wrote several small Python applications to gain experience with the WaveDash controller before moving on to phase II.

3.2 Phase II - Learning wxPython

Once familiar with how to use the WaveDash controller, phase II consisted of learning wxPython, a GUI framework for Python. The student spent about a month learning the framework and making mock-up GUIs for the project.

3.3 Phase III - Final Development

The final phase consisted of bringing together knowledge of both the WaveDash controller and wxPython to create a GUI application capable of running and controlling the waveforms the student had previously created. The final design, shown in figures 5 and 6, first allows the user to select the AM or Narrowband FM radio. In either case, the output volume can be controlled with the top slider which controls gain on the amplifier between the demodulator and the sound card. When the AM modulation type is chosen, the user can specify the frequency to listen on and can adjust the squelch to control the amount of static. When the narrowband FM is chosen instead, the user first has to decide whether to transmit or receive. Then he or she can enter a frequency. The Start/Stop button installs and uninstalls the waveforms, and the reset button resets all of the controls to their default states.

All of the controls that alter radio parameters work by using the WaveDash controller to configure properties of the waveform components. Configuring the properties is a fairly complex operation, involving several CORBA method calls. The WaveDash controller hides most of this complexity however, allowing the user to simply specify a component, a property, and a new value for the property. Installing and uninstalling the waveforms is similarly complex, but again the WaveDash controller hides the complexity. The user only has to specify the name of the waveform he or she wishes to install or uninstall.

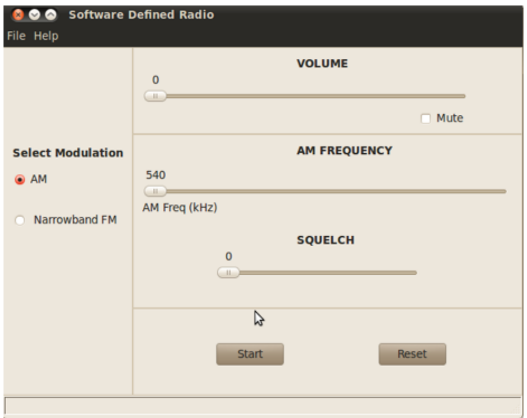


Fig. 5: The Final GUI: AM Radio

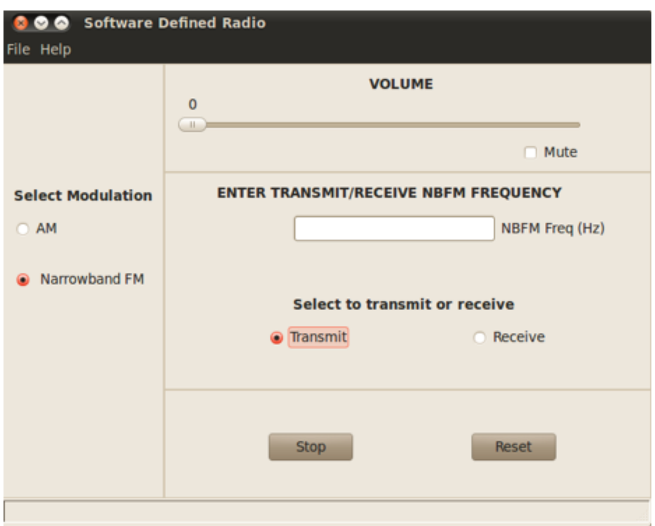


Fig. 6: The Final GUI: FM Radio

4. Measuring SDR Performance

After completing the final application, the FM transceiver was analyzed to determine the minimum signal strength necessary to produce detectable audio. Testing was accomplished using a combination of ALF, WaveDash, a function generator and an oscilloscope. Once the waveforms were installed by OEF, either ALF or WaveDash could be used to install and start them. ALF was then used to plot the output of either waveform while WaveDash was used simultaneously to change property values while the waveform ran. The plot from ALF could then be observed to make sure it changed appropriately. During testing, the signal power level into the USRP was adjusted from about -95 dBm to -40 dBm while observing the spectrum of the received signal and monitoring the audio out of the demodulator to determine the power level at which a discernable signal could be demodulated. The spectrum plots generated using ALF enabled comparison of observed signal bandwidth with bandwidth calculated using Carson's Rule [7]. In the case of the AM signal, both the carrier and modulating frequency were easily observable.

5. Conclusion

Initially released in 2004, OSSIE has been used in the classroom both at Virginia Tech and at the Naval Postgraduate School. The set of included labs teach users how to use OSSIE by engaging them in active learning. The Waveform Workshop abstracts away many of the details of the SCA, greatly reducing the learning curve involved with SDR research. The student who completed the project described here had no prior experience with the SCA or SDRs. Despite that, she was able to design waveforms and implement and test two waveforms, along with a GUI application to control them, in just over two months. At the same time, those who possess more in depth SDR knowledge can use OSSIE as well. It has also been used in a variety of research projects as described earlier. By combining easy to use tools with a robust core framework, OSSIE makes an ideal platform for SDR education and research.

References

[1] Corba faq. http://www.omg.org/gettingstarted/corbafaq.htm.
[2] Ettus research. http://www.ettus.com.
[3] Ossie project web site. http://ossie.wireless.vt.edu.
[4] Software communications architecture: Home page. http://sca.jpeojtrs.mil/.
[5] G. Abgrall, F. Le Roy, J.-P. Diguet, G. Gogniat, and J.-P. Delahaye. Predictibility of inter-component latency in a software communications architecture operating environment. In *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1 –8, 2010.
[6] C. Ahn, J. Kim, and S. Choi. Implementation of smart antenna and transceiver api on ossie platform for wireless innovation forum standards. In *SDR Forum Technical Conference*, Washington DC, December 2010.
[7] L. W. Couch. *Digital and analog communication systems*, page 324. Prentice Hall, 6th edition, 2001.
[8] J. A. DePriest. A practical approach to rapid prototyping of sca waveforms. Master's thesis, Virginia Tech, Blacksburg, VA, April 2006.
[9] C. Gonzalez, C. Dietrich, S. Sayed, H. Volos, J. Gaeddert, P. Robert, J. Reed, and F. Kragh. Open-source sca-based core framework and rapid development tools enable software-defined radio education and research. *Communications Magazine, IEEE*, 47(10):48 –55, 2009.
[10] E. Grayver, H. Green, and J. Roberson. Sdrphy - xml description for sdr physical layer. In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pages 1140 –1146, November 2010.

[11] B. Gu, J. Jung, K. Kim, J. Heo, N. Park, G. Jeon, and Y. Cho. Swicom: An sdr-based wireless communication gateway for vehicles. *Vehicular Technology, IEEE Transactions on*, 59(4):1593 –1605, May 2010.

[12] S. Hasan, P. Balister, K. Lee, J. Reed, and S. Ellingson. A low cost multi-band/multi-mode radio for public safety. In *SDR Forum Technical Conference*, Orlando, FL, 2006.

[13] E. Redding, T. Rittenbach, H. Satake, and C. Dietrich. Ossie-based gra testbed. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1 –8, 2009.

[14] M. Robert, S. Sayed, C. Aguayo, R. Menon, K. Channak, C. V. Valk, C. Neely, T. Tsou, J. Mandeville, and J. H. Reed. Ossie: Open source sca for researchers. In *SDR Forum Technical Conference*, 2004.

[15] D. Seeralan, S. H. Edwards, and C. B. Dietrich. The waveform dashboard: An interactively configurable gui for prototype sca-based sdr waveforms. In *SDR Forum Technical Conference*, Washington DC, December 1-4 2009.

[16] S. Singh, M. Adrat, and M. Antweiler. Nato rto/ist rtg on sdr: Demonstrating portability and interoperability of sca-based waveforms. In *SDR 09 Technical Conference and Product Exposition*, December 1-4 2009.

[17] J. Snyder, S. H. Edwards, and C. B. Dietrich. Towards an open-source integrated development environment for sca-based components and waveforms. In *SDR Forum Technical Conference*, Washington DC, October 2008.

[18] T. Ulversøy and J. O. Neset. On workload in and sca-based system, with varying component and data packet sizes. In *RTO Information Systems Technology Panel (IST) Symposium*, Prague, Czech Republic, 2008.

Preparing Software Engineering Graduates For IT 2.0 Era

Jie Liu[1] **and Feng Liu**[2]
[1]Department of Computer Science, Western Oregon University, Monmouth, OR, USA
[2] School of Computer Science, Beijing Jiaotong University, Beijing, China

Abstract - *Computer Science is a fast evolving discipline. As a result, we have to constantly improve our curriculum to response and anticipate changes in the field. This paper discusses what we have been doing and what will be doing to prepare our students to be more employable and be able to contribute to the growth of the organizations they will be part of.*

Students benefits from the many measures that have been in place for several years. The measures include soliciting inputs from industrial leaders to identify gaps between our teaching and industry needs while emphasizing on education, having our students interacting with field engineers by bringing them to school to enlighten our students about what does it mean to be a software engineer, developing applications that abstract from real world problems provided by real clients while expending solutions to answer a large set of similar situations, etc.

Soon, our students will develop and deploy their application in the Cloud, develop and consume web services, develop software following the SaaS model, learn about security by looking into existing components that handle authentication and authorization. In addition, students are learning parallel programming concepts, writing code for multi-core computers, and developing and implementing parallel algorithms that capable of bringing CPU utilization to 100%. We are also developing courses to introduce students developing applications for smart phones.

Keywords:

1 Introduction

If Amazon, Microsoft, and Google could have their ways, EC2 (Amazon), Azure (Microsoft), and AppEngine (Google) would soon replace millions of servers and other IT infrastructures in many enterprises worldwide. If that would have happened today, companies should have their IT departments supporting the same business computing capabilities largely through SaaS based software running on some virtual servers in the Cloud just to tap into this new “economies of scale [1]” wave while, in theory, saving a large portion of their IT budget on equipment and engineering hours [5]. In his recent book "Business/IT Fusion," Peter Hinssen, one of Europe's most sought-after leaders on the impact of technology on society and business, indicated that we are at a tipping point for transitioning from IT 1.0, where IT being a supplier for businesses, to IT 2.0 where IT is completely integrated into business [2].

Regardless how we label the new directions of IT, one thing is inevitable--we are steadily moving toward commoditization of hardware and software and depending more and more on the Internet and other communication devices. Now the question we need to answer quickly is: as a higher education institution, how can we, Western Oregon University (WOU) and Beijing Jiaotong University (BJU) as examples, prepare our students to be ready and employable in the near future? We do believe that many other universities worldwide are asking the same question.

In this short paper, we will discuss several effective measures that are in place now at WOU. While these measures are continuously carried out, we are introducing a few new actions to ensure that our students are well qualified to compete for openings and are capable to contribute to the growth of organizations employing them.

2 Several key measures contributing to our program's success

Most students exiting from our programs start their career immediately. Our students have been successful in securing jobs and performed well on their jobs. For example, in 1997, 1/3 of our WOU's graduates were hired by Intel at Portlan Oregon site. One of them, stood out among thousands engineers and researchers who have Ph.D. degrees and/or are graduates from legendary universities worldwide, received the Employee of the Year award from Intel a few years after.

To help students to be successful in their job search and career development, we introduced several measures that emphasize on learning by doing and learning by solving real world problems. Fundamentally, we believe the "teaching someone to fish" principle in our programs. This thinking is reflected in many measures that include soliciting inputs from industrial leaders to identify gaps between our teaching and industry needs while emphasizing education, having students interacting with real engineers by (1) bringing field engineers to school and (2) send them to observe operations in the field to enlighten our students about want it means to be a software engineer in the real world, developing applications that abstract from real world problems provided by real clients while expending solutions to answer a large set of similar situations. We believe that these measures are among the key elements contributing to our students' success and are still effective for preparing students entering job market in the IT 2.0 era.

2.1 Incorporating an industrial council

As professors, we may be strong in the theoretical aspect of subject areas of our interests; however, we recognize that we may not be current in prescribing the skill set of some special areas and may not up to date on personal characteristics the industry currently expecting. So, we formed an industrial advisory council as a pathway to gain help from industrial leaders from small non-profit organizations to large international businesses to local and state government agencies. We have been incorporated the suggestions to many of our courses and curriculum activities.

During our meetings, we always ask them to describe their demands, expectations and qualifications to our students. We have heard many things. For example, suggestions for students to be better prepared to support the IT industry are resume preparation, presentation skills, interview skills, communication skills, the ability to learn, and internships, etc. Interestingly, a strong Computer Science foundation was not ranked high because they assume that we are doing a good job educating our students with fundamentals.

2.2 Building team player characters

Knowing that very few sophisticated systems were built by a single engineer, we want our students to be team-plays. We know that students often work in collaborations during their school experience among friends. However, students lack the experience of working on a single project for an extended period, especially with others of not their choosing. This "working with others of not their choosing" is critical in software development process. Our approach, in part inspired by the industry advisory council, toward incorporating this experience is to assign each student into a team of three or four students in our two-term software engineering sequence. The selection process is largely driven by the students' past academic achievement in Computer Science coursework. Formation of team does not change except in the situation of student withdrawing from the classes.

The team functions like a small company that is managed and guided by the professor/manager. The team selects a name, creates a logo, business cards, web site, and even makes business stationery. The team members rotate to be the team leaders for individual milestones and then coordinate the administrative activities such as scheduling meetings and making sure all materials are completed and correctly formatted.

This approach creates a very competitive scenario. The teams with highly motivated and capable students raise the bar for the entire class, illustrating to their classmates what is possible. The teams with relatively weak students learn the real-life lesson of the workplace, unlike a class assignment where everyone is working on solving the same set of problems, a person can be the only one working on a specific task. Now, they must attempt and perform, or founder and fail. We have seen many cases where teams with supposedly “less able” students are challenged and inspired, they become productive.

2.3 Dealing with real clients

Immediately after the formation of the teams, each team has one week to solicit a client, usually a local company with some real world projects. The client provides the project requirements and domain knowledge from which the students must develop use cases. The teams learn firsthand that it is a mission impossible to create accurate and stable system requirements. We believe our students benefit from working with off-campus clients. The experience of solving a real problem for a real client also broadened students experience and help in building students’ confidence level. At the same time, we actually help some small business, government agencies, and non-profit organizations automated some of their problems. The students also benefit for this process because potential employers view this unpaid “working experience” positively.

The true success of the project is not only in the resulting software, but in what can be learned from the process. Students understand important concepts such as “learning on the job”. The drag-and-drop environment of today's IDEs is extremely powerful and can be learned with effort and persistency.

2.4 Learning from the real players

Knowing students can always benefit from the industrial experts, we invite many guest speakers from industry to talk to our seniors. The speakers include CEOs, owners of local software development firms, senior managers, software engineers (most of them are our former students who went through our program), and some independent consultants.

Understandably, many speakers echo the same message. For example, several speakers emphasized that a good software engineer is one that gets things done on time under pressure. The message must have sunk in because generally we have fewer students gave excuses why they did not have their tasks finished after such talks. Whenever possible, the visitors critique students’ resumes, perform mock interviews, and share inside tricks on answering tough questions. Some collect resumes to bring back to their organizations.

Many of BJU students will work for the railroad industry. Before graduation, we will bring them to large train station to observe operations in the field. Students are encouraged to find internships to further strength the knowledge acquired in classroom.

For students interested in conducting research, they are encouraged to join professors' research groups. Many students are co-authors to published works, and the experiences and skill in problem solving can be very beneficial in the students' professional career.

3 New measures targeting IT 2.0

In our view, the IT industry's marching toward commoditization of hardware and software affects three groups of engineers: first, the group of engineers who build and maintain the necessary grids and data centers, second, the group of engineers who architect and develop software running on the Cloud, and third, the group of engineers who support an enterprise's computing needs. Knowing our students need to be able to build a successful career in any one of these three groups, we have added or plan to add additional measures to better prepare our students to be successful in the IT 2.0 era, however, most addressing the needs for the second and third groups. We present them here hoping to contribute the discussions on the same subject and will continuously search for more ideas that help our students to learn, grow, and be successful.

Soon, our students will develop and deploy their application in the Cloud, develop and consume web services, develop software following the SaaS model or other service models, learning about security issues, such as authentication and authorization, by studying code that available through the open source communities. In addition, students are learning parallel programming concepts, writing code for multi-core computers, and developing and implementing parallel algorithms that easily bring CPU utilization to 100%. We are also developing courses to introduce students developing applications for smart phones.

3.1 Gaining understanding over designing and developing software for IT 2.0

It is impossible to know all technologies that will re-surface or be introduced to support developing software products for the not too distant future. However, a few things are most like to be true: a large number of software products will run on some virtual servers in the Cloud, will use web services, will most likely employ concurrency and distributed algorithms, and will need to support much strong security. If we do not drastically change our curricula, our students will not be ready to handle the challenges in front of them and may even miss many career opportunities. We do believe that the Cloud is the future for IT. The same believe is shared by many. In one article the author showed that in 18 month starting that from "January 2008, the growth in job postings that mention cloud computing has hit 350,000%" [9].

Programming tools such as Visual Studio and Netbeans have largely improved the developers' productivity, so projects took our students two months to build during the .COM era would only take current students a few weeks. However, we take the opportunities to add many new requirements that touch the fundamental building blocks we believe will be widely used in developing software products to come.

3.1.1 Using of virtual servers

For many years we have been requiring our students to carry out their software development activities on a server we maintain. Students gain access to the shared machines through remote desk connection utilities. Initially, the main reason for adopting this approach was costs. Then we realized that with shared servers we can control the development environment and monitor student activities and progress easily. In addition, students do not have to struggle with obtaining the necessary software and set up a working environment.

We plan to take this approach to development and deployment on the Cloud. We will be requesting funding to actually rent a few servers from Amazon EC2 or Microsoft's Azure so students can experience software development in the Cloud and learn to consider additional factors such as pricing structure, availability of software packages, maintaining software remotely, fault tolerance, securing customer data in the Cloud, and selecting and using of monitoring packages.

We believe that the experience of developing software products in the cloud computing environment is one of the best way to introduce our graduating seniors about cloud computing. If the bandwidth and funding allows, we may even allow our sophomores and juniors to write a few program on these rented machines.

3.1.2 Using of web services and SaaS concept

Web services are building blocks for SaaS based systems and SOA. They are an important building block as evident by their wide use in Amazon's EC2. Our students need to have a solid understanding about the concept and technology. So, for a couple of years, we have been requiring our graduating seniors to create a few web services and to consume a few, created by teams outside of our campus, as part of their senior projects. Their experience in producing and consuming of web services have fundamentally changed their view about software development and distributed computing.

Coding web services also provides an excellent opportunity for students to re-learn some of the important concepts such as security, state management, and object serialization/de-serialization. Student also learned about the power of standardizations.

A new requirement introduced recently is that students have to generalize their project to incorporate the SaaS model. For example, if a group of students' project is to develop a web-based inventory control application for a local drug store, the team then needs to build a SaaS application to support the similar operations for small stores or small drug stores in general. This requirement adds a new dimension to their senior project and greatly increases the complexity of their projects. It also forces students to learn many concepts detailed in SaaS and other service models.

Combining with developing and deploying their application in the Cloud, the students gain firsthand knowledge regarding Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), and Everything-as-a-Service (EaaS).

3.1.3 Introducing of parallel processing

For many years, we taught parallel programming class as if students would use the knowledge to write a few very specialized applications. Recently, multi-core computers become the norm, not to mention the number of cores come with a CPU is increasing steadily. Lately, we have redesigned our parallel programming classes to explore the benefit of concurrency through multi-core.

Besides learning new algorithms, reviewing many concepts, and watching the CPU's utilization reaching 100% and stays there, students enrolled in the parallel programming class truly have experienced "thinking outside the box" many times with each of the new algorithms introduced. It is like that we all of a sudden dropped a curtain to let the students see a new dimension because up to this point, students have been trained to come up with a sequential approach to solve just about any problem.

3.1.4 Learning to appreciate the complexity about security issues

When asked, every student knows the importance of security. After the requirement of adding a role based user management components to their project, students start to appreciate the fact that other people have already implemented and tested so many algorithms such as encryption and hashing. Incorporating such a component into their own application provides students an excellent opportunity to learn how experienced software engineers write their code. It also allows student to appreciate the usefulness of recent concepts such as Open ID.

3.2 Supporting an enterprise's business computing needs in IT 2.0

For software engineers developing business software integrating with outside systems, the skill set discussed in previous section still applies. To train engineers, architects, and managers making decisions on whether to use, say, a SaaS vendor, or which vendor to select, we believe they have to sharpen their skills on the job. Nevertheless, it is possible to introduce them some of the key considerations and skills such as the calculation of TCO and to understand a SLA from a SaaS vendor.

We plan to develop a graduate level course together with faculty members in our Business division and prelaw program to look into these issues because we believe that such a course needs to cover considerations involving technologies, business needs, and law and regulations. The interdisciplinary nature determines that such a course must be taught by experts of their own field.

3.3 Developing of enterprise level software

To different people, the definition of enterprise level software can be very different. Still, most would agree the software we developed in school is far from being considered as enterprise level software.

We will soon make available another class that covers some of the important characteristics for software designed to be deployed to support enterprise operations. The class will have students develop relative large software solutions with groups of students using agile software development methodology. Students must use version control, issues tracking, configuration management, and logging of executions. The students will also need to submit they code with unit tests. Project deployment will follow the practice of Continuous Integration

Another key characteristic is that this system developed by our students must have a high level of fault tolerance that comes from two directions: (1) always assumes that an operation, such as database connection, can fail so retrying of every operation is build in, and (2) employee monitoring tools into the system so the software support self healing. One system test these systems much pass is that if we shut down the database server and then bring it back, the system cannot enter an uncertain state when the database server is not available, and be able to reconnect to the database server when the server is back.

We are confident that students signing up the class are the ones really enjoying the software engineering aspects of software development, and the class will make them much better software engineers.

4 Conclusions

More and more students will be facing a working environment where commoditization of hardware and software are common. This short paper discussed several measures we have been using and we believe will still apply. We listed a few new measures that we believe will better prepare our students to be successful in the new IT era and hope to stir up discussions on ideas to achieve similar goals.

5 Acknowledgement

We would like to thank the Division of Computer Science at Western Oregon University, especially Dr. Marsaglia and Dr. Morse, and Oregon Engineering and Technology Industry Council for their continuous support of our research.

6 References

[1] M.J. Miller, "Cloud Thinking: Amazon, Microsoft, and Google", PCMag.Com Blogs, November 12, 2008

[2] P. Hinssen, Business/IT Fusion, Mach Media NV/SA, London, December 16, 2008

[3] J. Liu, J. Marsaglia, and D. Olson, "Aiming for Success after Graduation", in the Proceedings of ICFE2005 June 20-23, 2005, Las Vegas, pp. 127-132

[4] http://www.microsoft.com/education/solutions/ cloudcomputing.aspx

[5] A. Fox, Cloud computing in education, http://inews. berkeley.edu/articles/Spring2009/cloud-computing, March 17, 2009

[6] D. Harrison, Is Cloud Computing a Credible Solution for Education?, http://campustechnology .com/articles/2009/11/12/is-cloud-computing-a-credible-solution-for-education.aspx, 11/12/2009

[7] D. Patterson, Cloud Computing and the RAID Lab, research.microsoft.com /en-us/events/cloudfutures2010

[8] IBM, Object-Oriented Technology Center, "Developing Object-oriented Software", Prentice Hall, Upper Saddle River, New Jersey, 1997

[9] D. Linthicum, Need a job in IT? The cloud needs you, http://www.infoworld.com/d/cloud-computing/ need-job-in-it-cloud-needs-you689?source=IFWNLE_ nlt_cloud_2009-12-07, December 03, 2009

Understanding Software Complexity Issues in Software Engineering Courses

Stuart Steele and Keni Yip
Polytechnic Institute of NYU, Department of Computer Science and Engineering,
6 Metro Tech Center, Brooklyn, NY 11211

Abstract - *The complexity issues related to modern software systems continue to grow and insights into key factors influencing complexity must be better understood. This paper indicates how complexity factors are investigated and understood with two graduate level software engineering courses. The effort includes hands on development and tool use in addition to review of extensive number of real world examples and a major project. The utilization of metrics is addressed and two major papers in the complexity area are analyzed.*

Keywords: Software Engineering, Software Complexity, Software Design, Projects, Metrics

1. Introduction

It is evident that software development is a complex process and can result in very complex designs and code. But how do we assess complexity? What can we do to minimize complex design and code? Certainly the requirements of the problem can be complex and all associated code may exhibit this complexity. Can the complexity of the design and code be validated utilizing metrics? What is system level complexity vs. code complexity? This paper addresses approaches and tools that have been used in a graduate software engineering sequence of two courses to address this complexity question. Software metrics is often a hot item for discussion but the focus here is on complexity: how we can educate the student to understand software complexity and how to deal with it. While the McCabe Complexity Metric still may be useful, it is really a small subset of complexity assessment especially in object oriented design. While many studies on software faults and failures have been conducted, the lack of publicly available real world fault and failure data and the lack of published studies conducted on such data have been recognized by many. (1) (2)

Many companies track failures in their own software, but there is little attention paid by the field as a whole to historic failures and what can be learned from them. Our graduate software engineering courses (two course sequence) contains four major efforts that relate to complexity. 1) The complexity concept, the study of complexity using some of the most recent and applicable papers to highlight the issues, 2) A project developed by a student team using a modern software engineering UML tool set. 3) An evaluation of many real case studies that experience major difficulty, in order to summarize and assess patterns and techniques, and 4) A major design project, stressing requirements and design.

What is the definition of complexity? Recently there has been an increased emphasis on using metrics, such as Chidamber and Kemerer (3) and others to understand and minimize problems with large scale software systems. While this is of interest, results do not seem to help in predicting faults in practice.

1.1 The View of the Problem and Solution:

The system and software view of projects also gives us different perspectives. For the famous Ariane 5 spacecraft problem, a system view gives one type of answer while the software design/code view is quite different. The fundamental issue here relates to whether this is a system level or software level problem, or a project or management problem. There are also tools that help in design and development that can support minimizing the complexity of the software solution. One such tool is the IBM Rational Software Architect and is used in the development project in the first software engineering course. This supports code to model and model to code transformation. Software metrics are also used on the project and include design and code level metrics: SLOC – Source line of code, WMC – weighted methods per class, CBO – Coupling between objects, RFC – Reference for Class, etc.

2. Complexity Concept: (First Course)

To understand complexity issues in software development and their impact, it is important to analyze results that have been documented in the recent literature. Students were asked to review and comment on two papers on software complexity and their relationship to a software development project. The papers address complexity of large software systems and both discuss common trends in fault and failure. (1) Common Trends in Software Fault and Failure Data, IEEE Transactions on Software Engineering, July/August 2009 by M. Hamill & K. Goxeva-Popstojanova, (2) Conquering complexity, IEEE Computer, December 2007 by G.J.Holzman.

The first paper discusses the definition of a fault and a failure, a very important concept. “A failure is a departure of the system or system component behavior from its required behavior. Failures represent problems that the users (human or computer) see.” Fault is an accidental condition that may cause the system to fail to perform as required. This is how past work has shown the results. However, not every fault corresponds to a failure since the conditions under which faults result in a failure may never be met. The results show that individual failures are often caused by multiple faults spread throughout the system. Also a large number of failures are linked to late life cycle activities. The four major types of problems found are defined as requirements, design, coding faults, and data problems.

The paper by Holtzman (2) is consistent with Hamil’s paper by indicating that the probability of any one fault can be low, but experience shows this often leads to major problems. He indicates that most software failures in space missions can be reasoned back to unanticipated combination of otherwise minor issues. The bottom line in complex systems is that combinations of minor software fault defects can lead to large system failures.

The students were required to answer the following 5 questions related to the papers shown in Table 1.

Table 1 – Paper Review Questions

Questions	**Answers**
Question 1- Do both papers indicate the same behavior of faults and failures? Explain.	Question 1 – Both papers indicated the same behavior of faults and failures. These results were different than previous studies that have claimed that the majority of failures were the result of single faults or localized faults.
Question 2 - Does the Common Trend paper answer the two main research questions: 1-Are faults that cause individual failures localized, that is, do they belong to the same file, or component. 2) Are some types of failures, more common than others. Explain.	Question 2 – 1) Based on extensive data gathering, paper #1 show that faults that cause individual failures are not often localized. 2) The requirements, code and data errors were most common.
Question 3 - Do you think collection of classical modern metrics, such as LOC, WMC, DIT, LCOM, CBO and RFC primarily related to class and sequence diagrams can give information to highlight and solve some of the problems listed in the above paper? Explain.	Question 3 – The basic metrics from Chidamber and Kemerer are useful for identifying problems but do not address all issues. These metrics should be emphasized with more examples.
Question 4 - What is your personal view on how to address the complexity of software systems in order to minimize problems? Explain.	Question 4 – Techniques varied, but many students really stressed the use of the common metrics. However these do not often map well with requirements, design and coding errors.
Question 5 - After reading these papers and you own hands on experience what is your assessment of the current state of the art in understanding complexity in software systems. Please circle one of the following – Excellent, Good, Average, Fair, Poor, and explain.	Question 5 – 47% of the students assessed the current state of the art on understanding complexity in software system as ***Fair*** or less while many students gave it too high a value. This appears because they were impressed in the data gathering effort and documentation of the NASA project. However many projects are not like this.

3. Project Example: (First Course)

It is possible to assess complexity of a student development project that allows complexity of the solution to be addressed on a couple of levels. Last semester the course project was given to 12 groups of four students, with one team leader in charge of each team. The software was to be developed using the IBM Rational Software Architect (4) using good object oriented principles. The language was selected by the team and was either C++ or Java. The development project was evaluated using modern development tools, with traceability from requirements to code and back again involving class diagrams, sequence diagrams and well structured code. Software metrics were applied later, to determine complexity factors and an assessment of overall errors.

The project development problem was to design and develop a basic game for demonstration in the Software Engineering Lab. The game had two levels of difficulty and had built in protection for hitting the wrong keys. Between games the logic was randomly changed. The teams were provided an overall state diagram and the requirements of the object design and development approach.

Typical project assessment items assessed at the demonstration.

- Overall Presentation
- Class Diagram
- Sequence Diagram
- Code/Comments
- Demo Level 1
- Demo Level 2
- Error Conditions
- Tool Set Integration

Table 2 is a User Demonstration Table that summaries the user type information presented at the lab demo. Presentation includes clarity and organization of the material, critical items in the approach, and any details that may be unique to the group's approach. Level 1 is one difficulty level of the game and Level 2 is more difficult. Errors refer to events that occur that should not occur, such as entering a random entry and the game not being protected. Also included is improper movement from level 1 to level 2. The results of the team performances are shown in table 2.

Table 2 – User Demonstration Grade

Team	Presentation	Level One	Level Two	Errors	Total Grade
1	A	A	A	A	A
2	B	A	C	C	C
3	B	A	C	A	B
4	A	A	A	C	B
5	B	A	A	A	A
6	B	A	A	A	A
7	A	A	A	A	A
8	B	A	A	A	A
9	A	A	A	A	A
10	C	A	A	C	C+
11	A	A	A	A	A
12	A	A	A	A	A

The design, code structure, code comments and tool integration were reviewed in detail by the laboratory student support team after the demonstrations. The results are in Table 3.

The results differ in tables 2 and 3 since the user table is only what the user sees, while the Technical table includes the details of the design and code. Many of the teams had teamwork problems. Some met often, others met once a week and carried out work independently and then integrated it together. Some teams had an individual named as test director. Some teams utilized remote development, integration and testing. Design includes the appropriate use of inheritance, aggregation, association and the use of controllers.

Metrics were evaluated after each team had demonstrated their project. The class diagram, sequence diagram and code were reviewed by the lab staff. The number of classes if low, indicated poor object oriented design, and if high, often lead to architecture level issues and poor class encapsulation. Table 4 shows the metric results that vary from 4 to 1 with 4 being best.

Table 3 – Overall Project Technical Results

Team	Design	Code	Comments	Tool Integration	Language	Total Grade
1	A	A	A	A	Java	A
2	B	B	C	C	Java	C
3	B	A	B	C	Java	B
4	A	A	A	A	Java	A
5	A	A	A	A	C++	A
6	A	A	B	B	C++	A-
7	A	A	A	A	C++	A
8	A	A	A	C	Java	B+
9	A	B	C	C	Java	B
10	B	A	C	B	Java	B
11	B	B	C	C	Java	B
12	B	A	C	C	Java	B

Table 4 – Metrics Assessment Grade of Project Teams

Team	LOC	WMC	CBO	LCOM	RFC	# of Classes	Total Grade
1	4	4	3	4	4	11	A
2	3	2	2	3	3	3	C
3	3	3	3	4	3	14	B
4	4	4	4	4	4	9	A
5	4	4	3	4	3	9	A
6	4	4	3	4	3	9	A
7	3	4	4	4	4	5	A
8	4	4	4	4	4	8	A
9	3	2	3	3	4	10	B
10	3	4	4	3	4	6	A
11	2	3	4	3	4	3	B-
12	3	4	4	4	4	18	A

The CBO and LOC may be the best metrics to use, since LCOM and WMC are most applicable to individual class complexity. These metrics have been evaluated for effectiveness and show promise for fault prediction (5). The teams with low number of classes did not have a good object design while teams with a high number of classes also had awkward architectures. Table 5 shows the combination grade from table 2, 3, and 4.

Table 5 – Total Grade, All Evaluations

Team	Technical Grade	Metric Grade	User Grade	Total Grade
1	A	A	A	A
2	C	C	C	C
3	B	B	B	B
4	A	A	B	A-
5	A	A	A	A
6	A-	A	A	A
7	A	A	A	A
8	B+	A	A	A-
9	B	B	A	B
10	B	A	C+	B
11	B	B-	A	B
12	B	A	A	A-

This project highlighted what is complexity in software systems. The class structure and the overall architecture impacted the complexity of the work as perceived by the students.

4. Case Studies (Second Course)

In the second software engineering course, we have the assessment of case studies. These come from all sources, such as real time and enterprise systems, and entail both management and detailed development problems. A major goal is to identify the "root cause" and then the overriding problem and situation that allowed this. Insight into the type of problems and cost is important. In addition complexity levels are identified and then analyzed to see what techniques could be used to minimize the risk and reduce complexity.

The question can then be asked, are any of these errors systemic and therefore common to many systems?

The projects have either development problems or latent problems which occur after sell off. 35 examples are latent type problems and 17 are development problems out of a total of 52 case studies. The projects are presented to the students summarized by issues and situations. A pattern of faults per system is shown in Table 6.

Table 6 – Pattern of Major Errors Found in Case Studies

Requirements	Design	Code	Data
20%	30%	40%	10%
Latent	**Development**	**Latent & Development**	
67%	33%	15%	

These projects are classified by four types of errors, Requirements, Design, Code and Data problems as indicated by the NASA experience of Hamill. When they occur is identified as either Latent or after delivery, or Development including all phases leading to unacceptable delivery. Some projects include both Latent and Development problems. Often there will be arguments that the real problems are management, not technical.

The major focus on these projects is to find the "root" cause, and then give a statement of why the problem occurred. Also, similar type problems are highlighted.

5. Major Design Project: (Second Course)

A major design project is required by teams of 4 students each. The emphasis here is on requirements tracing and good design approaches. A student team design is reviewed by every other team and the overall complexity of the approach is assessed. The system to design is an Electronic Voting Machine (6). A major item is mapping between user requirements and top level design and architecture. The Major focus is on security elements and creating a complete specification of all critical features of the voting software. The team specification document includes the following elements.

5.1 Elements for Project Evaluation:

- Allocation to Subsystems and Software Architecture
- Functional Requirements and Features – Map to RFP
- Major Use Cases
- Hardware/Software Environment – Block Diagram
- Analysis Level Class Diagrams
- Sequence Diagrams (Major)
- Formal Methods Specification
- Proposed Development Methodology
- Solution to Eliminate Over Votes and Under Votes
- System Complexity Assessment
- Readability and Organization of Document
- Risk Management Approach

Each team report is evaluated by the other teams and by the instructor. Differences in evaluation by different teams are evaluated by the instructor. Each member of a teams' work is reviewed by every other member of the team, and address percent participation and competence level (A, B, C, and F).

6. Summary

As one develops modern systems that are large and complex it is important to understand elements of complexity that can influence the development of a successful software intensive system. Emphasizing complexity and the importance for a quality product, forces a focus on how to minimize issues that will adversely affect the product. The four focus areas of this paper, Complexity Concept, Project Development with Modern Tools, Review of Case Studies, and a Major Design Project highlight this. These elements are covered in two software engineering courses. The bottom line is that large scale complex software systems will remain a challenge and the four focus areas give the students some very good experience and guidance into the issues. Student feedback has been extremely positive.

7. References:

[1] M. Hamill, K. Goseva-Popstojanova, "Common Trends in Software Fault and Failure Data",
IEEE Transacitons ofnSoftware Engineering, vol. 35, no 4. July/ August 2009.
[2] G. J. Holtzmann**, "**Conquering Complexity**", *IEEE Computer*,** December 2007.
[3] Chidamber, S. R., and Kemerer, C.F., "A Metric Suite for Object Oriented Design", ***IEEE Transactions in Software Engineering,*** Vol SE – 20, no. 6 June 1994.
[4] IBM Rational Software Architect Manual 2011.
[5] Gyimothy T., Ferebce R., and Siket T. "Empirical Validation of Object Oriented Metrics on Open Source Software for Fault Prediction" ***IEEE Transactions on Software Engineering,*** October 2005.
[6] Electronic Voting Machines – DRE – State of Connecticut RFP 2006.

Design Principles for a Beginning Programming Language

John T Minor and Laxmi P Gewali
School of Computer Science
University of Nevada, Las Vegas

Abstract: *We consider the issue of designing an appropriate programming language for teaching beginning computer science courses. We examine the features of widely used teaching languages that include C++/C#/Java and show that the constructs of these languages can be significantly improved to make them more effective for class-room teaching. We then propose a set of design principles that can lead to a pedagogical-friendly programming language with promising scope for enhanced learning .*

Keywords: programming language, pedagogical issues, language design principles

1. Introduction

In this paper we examine the issues of designing a programming language that can be adopted for undergraduate teaching. We propose a set of principles that can be followed as guide-lines for developing a student-friendly programming language. These principles have been formulated so that the resulting programming language can be used by beginners in the first courses on undergraduate programming. The widely used programming languages that include C/C++/C# and Java are not well designed for beginners. To meet the requirement of fast execution time, these programming languages are not user-friendly for fresh programmers. The widely used C#/C++/Java languages are complicated and can be improved for class-room and laboratory instruction. We examine some of the draw-backs of these languages and suggest approaches for making improvements. We show that some of the complicated programming constructs in C#/C++/Java are not needed at all in the first courses in programming.

2. Keep the size of the language small

A "teaching programming language" should have a small size in the number of reserved words, operators, statement/declaration constructs, and precedence rules. Unlike C++ for example, the language should be designed with less than 18 precedence levels. A number of features in C++/C#/Java can easily be eliminated because they are redundant or rarely used. In fact, the "struct" construction in C++ is redundant. Any situation where "struct" is needed, one can use simple class construction. C++ and Java allow comments to be written in two ways: (i) by using slash-star pairs (/* Comment */) or (ii) slash-slash (//Comment). The comment in the form /*…..*/ is not necessary and redundant. We can similarly examine the need for static variables. Static variables are rarely used in the first courses on programming. Static variables lead to debugging complications and can be eliminated without compromising expressive needs. The inheritance mechanism in C++ can be improved for instructional purpose. Single-inheritance is adequate for writing programs in beginning courses and hence multiple-inheritance is unnecessary. Programs that do not use multiple inheritance are easily communicable. Qualifiers such as "friend" or "protected" inside class definitions confuse fresh learners and should be addressed only later. Java uses a lot of "Interfaces" even in the first course. Interface-abstraction is not easily comprehensible to beginning students and its use is not necessary. Similarly, function parameters are rarely used and not needed.

3. Natural and consistent syntax

A student-friendly programming language should be internally consistent with minimal exceptions and be as natural as possible. The term "internally consistent" is used to mean that the language construct should not allow too many variations in the use of statement constructs. For example C++/C# does not make it mandatory to initialize variables. This freedom leads to errors that are hard to spot. A pedagogical friendly programming language should force the programmer to initialize every variable during its declaration. If a variable is not initialized when declared then the compiler should flag an error. Proper initialization during declarations should be done for scalar and non-scalar objects as well. When an array is declared, values at all positions should be initialized. Furthermore, for array initialization, there should be a simple way to initialize all array entries with the same value. Even FORTRAN has simple aggregate array initialization. For objects, initialization by listing values for members should be made available to programmers. The way C++ does this by using "initializing list" is too cryptic and complicated. C++/C# have no mechanism for initializing the members' values, a source of inconsistency. One is compelled to use constructors, which is not consistent with simple variable initialization.

A pedagogical friendly language should allow user-defined functions to return **any** built-in primitive or structured type. In C++, arrays cannot be returned explicitly. They are returned in C++ either by wrapping in an object or by returning a pointer to the beginning of the array. This is confusing to beginners. It is natural that functions be able to return the array type directly with the size of the array explicitly known. Furthermore, assignment between array types and between structure types should be allowed. In C++/Java, assignments between objects are allowed but not between arrays; this is inconsistent. It is remarked that even the earlier versions of FORTRAN allowed assignment between arrays of the same type. The association rules of widely used programming languages are complicated for beginners. In C++, some associations are left-to-right and others are right-to-left, even for binary operators. It is necessary pedagogically to simplify association rules. One suggestion would be to make all binary operators have left-to-right association while all unary operators have right-to-left association. There should be no exception to these association rules.

Consider the issue of unnatural conventions used in the programming languages C++/Java/C#. One source of unnatural convention is that functions by default are recursive. Beginners find it hard to learn recursive functions. Recursive functions cannot be comprehended in a natural way in the first course on programming. It is more appropriate to make functions non-recursive by default. When recursive functions are to be used, they must be specified by being tagged "recursive". This will emphasize the fact that special care should be done while designing recursive functions. Another example of an unnatural convention in the teaching of programming languages is the placement of the return type in function definitions. In C++/Java/C#, the return type is written before the function name and parameter listing as in "*char max(char a, char b, char c)*". It is more natural to put the return type at the end, as in "*max(char a, char b, char c)-> char*".

The specification of parameter passing in function calls in the existing popular programming languages is unnatural and confusing. Take the example of parameter passing in C++: "*void findMax(int a, int b, int & c){..}*". In this example, the first two parameters *a* and *b* are passed by value and the third one *c* is passed by reference. Using the symbol '&' to indicate pass by reference is very cryptic and counter intuitive. The syntax of parameter passing should reflect intended purpose. It is much more appropriate to use tags "*in/out/in-out*" in parameter passing. Here, the tag "in" is used to indicate the parameter whose value is read for computation inside the function and the tag "in-out" is used to indicate two way communication. With this convention the findMax function can be written as:

void findMax(int a in, int b in, int c out){..}

Such a construct emphasizes, in clean terms, that the first two parameters are passed read-only and the third one is used to implicitly return the computed value. Chances of having unintended errors will be reduced by this convention.

4. No error-prone operators

A numbers of operators in C++/Java are very confusing when someone is learning programming. We can take the example of the increment operator ("++"). This operator is mostly used for increasing the value of an operand by 1. It is noted that the operand is incremented by 1 when operator ++ is applied as prefix or postfix. The main difference in the meaning of postfix and prefix increment operator is in the value of the expression *++i* or *i++*. In the prefix application the value of *++i* is the new value of *i* and in the postfix application the value of *i++* is the old value of *i*. We can imagine how confused a beginner would be when they have to get buried in such unnecessary detail when they are just starting to comprehend the meaning of elementary statements. The same thing can be argued for the decrement operator ("--") and extended assignment operators such as "+=", -=, *=, etc. It is critical pedagogically to replace unintuitive "i++" by the very intuitive assignment statement "*i = i+1*". Similarly, cryptic extended assignment "*x += 3*" should be replaced by simple assignment statement "*x = x+3*". These observances underscore the need for removing complicated side-effect generating operators that include ++, --, =+, =-, and =*. The pointer arithmetic operations used in C/C++ are very difficult to understand for fresh programmers. Errors due to pointer misuse are very difficult to debug. Java corrected this demerit of C/C++ by not allowing pointers at all.

5. Only high-level built-in types

There should be only a small-set of built-in types and these built-in types should be all high-level. The C#/C++/Java programming languages distinguishes primitive types into too many cases: integer, short, long, unsigned, etc. This forces the programmer to think about the underlying hardware while writing and analyzing programs. This slows down the learning process. The programmer should be freed from worrying about the low level hardware detail by leaving the issue of hardware choice to the compiler. Let the compiler select the best hardware representation based on the use as stated by the programmer. An appropriate way would be for the programmer to specify the range of an integer variable by writing down its lower and upper bounds. For example, consider a declaration such as "*int low … high m*", where this indicates that the integer variable *m* can take integer values between the limits *low* and *high*. The compiler then can translate to the appropriate hardware word-size. This eliminates the need of distinguishing between *short, integer, unsigned* and *long* for the programmer. This argument equally applies to float types etc.

The low level pointer-oriented memory allocation of array and structures practiced in C#/C++ are not easily communicable and prone to errors. Such constructs should be replaced with high level dynamic string and/or list types. Dynamic allocation of space should be done automatically when needed by string or list operations. It is remarked that de-allocation of unused memory should also be done automatically by a garbage collection technique and programmers should not have to worry about this problem (or the dangling reference problem).

6. Proper naming convention

There should be enforced fixed naming conventions for user-defined identifiers so that anyone scanning or debugging code will immediately recognize the purpose and role of an identifier from its form. A program written without these conventions should be flagged by the compiler. Typical naming convention could include: (i) Class names and enum types must start with a capital letter, and the rest must be capitals, digits, or underscores. (ii) User-defined functions or methods must start with a capital letter and the rest must be lower case letters, digits, or underscores. (iii) User defined constants, including **const** and **enum** constants, must start with a lower-case letter and the rest must be lower-case letters, digits, or underscores. (iv) User defined type parameters as used in generic class or functions, must start with a question mark (?)and the rest must be capital letters, digits, or underscores. (v) User defined variables including non-type parameters, objects and data-fields must start with a question mark (?) and the rest must be lower-case letters, digits or underscores.

7. Discussion

We outlined only a few principles for designing a teaching programming language. Due to space limitations we did not present a complete list. Some additional principles that would be useful in designing a pedagogical programming language could be issues dealing with selection of programming paradigm, side-effect, implicit operands, parametric polymorphism, debugging aids and dynamic error-checking. For example, the language should be flexible and not tied down to a particular programming paradigm. One-paradigm languages (e.g. Java) lead to awkward/complex implementations of those algorithms that do not "fit" the given model [1,4].

When functions and class definitions are first presented to the students, generic type parameters should also be introduced. Students should be taught that when functions/classes are written as general as possible, reuse is more likely, and that is desirable. Templates should NOT be a separate construct introduced independently late in the instruction.

Whether to (or not to) do dynamic error-checking should be at the option of the programmer and not decided by the language. Error-recognition capabilities (needed during project development) must be balanced with run-time efficiency (desirable in the final product). Both are useful at different phases of software development. A complete syntax and specification of a pedagogical programming language based on the principles discussed in this paper is available as a technical report [1].

References

[1] John T. Minor, "HIGH-C: A Pedagogical Language Based on High-Level Design Principles," *Technical Report* (20 pages) CRS-11-001, School of Computer Science, University of Nevada, Las Vegas, 2011.

[2] John T. Minor and Laxmi Gewali, "Pedagogical Issues in Programming Languages," *Proceedings of the IEEE International Conference on Information Technology*, pp. 562-565, April 2004.

[3] Daconta, Michael, Kevin Smith, Donald Avondiolo, and Clay Richardson, *More Java Pitfalls*, J. Wiley and Sons, 2003.

[4] Gewali, Laxmi and John T. Minor, "Multi-Paradigm Approach to Teaching Programming," *Proceeding of 2006 International Conference on Frontiers in Educations: Computer Science*, pages 141-146, June 2006.

[5] Daconta, Michael, Eric Monk, J. Paul Keller, and Keith Bohnenberger, *Java Pitfalls*, J. Wiley and Sons, 2000.

Learning Styles and Student Performance in Java Programming Courses

Chiu-Liang Chen and **Janet Mei-Chuen Lin**
Graduate Institute of Information and Computer Education,
National Taiwan Normal University, Taipei, Taiwan

Abstract - *It is crucial for a teacher to take students' learning styles into account to enhance teaching effectiveness. This study examines correlations between students' learning styles and their performance in writing and debugging Java programs. Eighty-four college freshmen who enrolled in two introductory Java programming courses participated in this study. Students were given the Soloman-Felder Index of Learning Styles test to determine their preferred learning styles, which were correlated with their performance on three tests—a written test, a hands-on programming test, and a debugging test. It was found that* sensing *learners performed the best on the written test, and the* reflective/sensing/visual/sequential *learners achieved the highest average scores on each of the three tests.*

Keywords: Java programming, Debugging, Felder-Silverman learning style model, Soloman-Felder ILS.

1 Introduction

The relationship between learning styles and learning performance in computer science courses has been investigated by many researchers. Allert [2] examined various factors, including students' learning styles, their previous computer experience, and their opinions of the value of programming projects and lectures, that might have contributed to successful outcomes in an introductory computer science course. He found that *reflective* and *verbal* learners outperformed *active* and *visual* ones; moreover, students' opinions of programming projects and lectures were highly correlated with their success in the course. Chamillard and Karolick [3] investigated how Group Embedded Figures Test, Felder's Index of Learning Styles, the Kolb Learning Styles Inventory II '85 and the Keirsey Temperament Sorter related to student performance in a computer science course. Their findings revealed that *reflective* learners performed significantly better than *active* learners on the Felder's *active/reflective* dimension. Similarly, Thomas, Ratcliffe, Woodbury, and Jarman [12] studied correlations between preferred learning styles and performance on both the exam and practical aspects of an introductory programming course. They found *reflective* learners scored significantly higher than *active* learners and *verbal* learners significantly higher than *visual* learners on the exam portion of the course. The study conducted by Pillay and Jugoo [8] revealed that in Kolb's Learning Style Inventory, *assimilators* outperformed *divergers* in a first course in procedural programming. Zualkernan, Allert, and Qadah [13] conducted a cross-cultural comparison of learning styles to examine the association between learning styles and performance in introductory programming courses. It was found that *reflective* students at two of the three universities achieved significantly higher grades than *active* students; students who were more *global* in their learning also performed significantly better.

Researchers in the afore-mentioned studies focused mainly on students' programming ability; seldom have we seen investigations into correlations between learning styles and students' debugging ability. Debugging is a critical component of the programming process. Fitzgerald et al. [7] pointed out that students who were good at debugging usually were also good at programming; conversely, students good at programming were not necessarily good at debugging. As such, many researchers (e.g., Ahmadzadeh, Elliman, and Higgins [1]) have suggested that more emphasis be placed on training students' debugging skills.

The aim of this study is to examine correlations between learning styles and programming ability in an introductory Java programming course. In particular, a debugging test has been included in the assessment of student performance so that correlations between learning styles and students' debugging ability can be investigated. The remainder of this paper is divided into four sections. Section 2 describes the Felder-Silverman Learning Style Model that is used in this study to determine students' learning styles; common mistakes frequently made by novice Java programmers are also discussed in the same section. Section 3 provides details of the experiment conducted in this research, Section 4 presents research findings, and Section 5 is the conclusion.

2 Related Work

2.1 The Felder-Silverman Learning Style Model

The Felder-Silverman learning style model was originally defined by Felder and Silverman [6] and later updated by Felder [5]. The purpose of this model was to capture the most important different ways in which students take in and process information. It had been used to help students to understand their own learning needs and provide a good basis for instructors to tailor their courses that meet the learning needs of all students.

The Felder-Silverman learning style model classifies students' learning styles as having preferences for one category or the other in each of the following five dimensions.

- *Active-Reflective.* Active learners tend to retain and comprehend information best by doing something active with the information—discussing it or testing it in some way or explaining it to others, whereas reflective learners learn best by thinking things out on their own.
- *Sensing-Intuitive.* Sensing learners tend to be more practical and careful than intuitive learners, and they prefer learning facts and solving problems with well-established methods. Conversely, intuitive learners tend to be more innovative than sensing learners and better at grasping new concepts.
- *Visual-Verbal.* Visual learners retain more from what they see—pictures, diagrams, flow charts, time lines, films, and demonstrations; in contrast, verbal learners learn better out of written word or spoken explanations.
- *Sequential-Global.* Sequential learners tend to learn in linear, logical steps; on the other hand, global learners prefer to learn in large jumps, absorb random pieces of material, and then suddenly "get it."
- *Inductive-Deductive.* Inductive learners prefer presentations that proceed from the specific to the general; contrarily, deductive learners prefer presentations that go in the opposite direction.

2.2 Error Types and Problem Types Encountered by Novice Java Programmers

Error types and Problem types encountered by novice Java programmers have been widely investigated. Ahmadzadeh, Elliman, and Higgins [1] collected and categorized all compiler-generated programming errors using a modified open source Java compiler, Jikes. It detected 108,652 errors from 15 exercises done by students in one semester. Sixty-three percent of the errors were identified as semantic errors, and the six most common semantics errors were: field not found, use of non-static variable inside the static method, type mismatch, using a non-initialized variable, method call with wrong arguments, and method name not found. Hristova, Misra, Rutter and Mercuri [4], in order to develop a pre-compiler that generates meaningful error messages to help novice Java programmers identify and correct program bugs before submitting their programs to the actual compiler, administered questionnaire surveys to college professors, teaching assistants, and students of computer science departments to gather data about Java programming errors. Among the 62 reported errors, 20 were identified as being essential errors that need to be dealt with by their educational tool, which included 13 syntax errors, one semantic error, and six logical errors. Robins, Haden, and Garner [9] investigated types of problems which often caused students to request help. The problems observed were recorded and classified with a problem list that comprised 28 pre-defined problems grouped into three types—background problems, general problems, and specific problems examined by pervious research.

3 Methodology

3.1 Participants

This study was participated by 84 college freshmen who enrolled in two introductory Java programming courses offered by the Department of Information Management at National Taipei College of Business. For most students the course was their first exposure to Java programming; however, many of them had had programming experience in Visual Basic.

3.2 Procedure

The experiment was conducted in Fall 2009. To determine students' learning styles, all participants were asked to fill out a Soloman-Felder Learning Style Index questionnaire at the beginning of the semester. During the 18 weeks of instruction that followed, students were offered a mixture of lectures and hands-on exercises which took place in a computer lab. There were three 50-minute class periods per week. Programming constructs covered included input/output, variables, arithmetic and logic expressions, classes and objects, data abstraction, encapsulation, conditional statements, loops, and arrays. The midterm and the final exams were administered in the 9th and the 17th week respectively. The exams were intended to assess students' comprehension of programming concepts, their ability to apply those concepts in programming tasks, and their debugging ability. Therefore, both exams included three parts: a written test, a hands-on programming test, and a debugging test.

3.3 Data Collection

In this section we describe the content of the Soloman-Felder Learning Style Index questionnaire which was used to collect data about students' learning styles; we also present examples of the problems given in the midterm and the final exams and how students' test scores were assigned.

- The Soloman-Felder Index of Learning Styles (ILS)

The Soloman-Felder ILS was developed by Soloman and Felder ([10]) based on the Felder-Silverman learning style model described in Section 2. It is an on-line instrument used to assess preferences on four dimensions: active/reflective, sensing/intuitive, visual/verbal, and sequential/global. The ILS questionnaire ([11]) consists of 44 questions, with 11 questions for each learning style dimension. A sample question for each of the four dimensions is shown below:

I more easily remember (a) something I have done. (b) something I have thought a lot about. (The Active/Reflective dimension)

I prefer courses that emphasize (a) concrete material (facts, data). (b) abstract material (concepts, theories). (The Sensing/Intuitive dimension)

When I think about what I did yesterday, I am most likely to get (a) a picture. (b) words. (The Visual/Verbal dimension)

It is more important to me that an instructor (a) lay out the material in clear sequential steps. (b) give me an overall picture and relate the material to other subjects. (The Sequential/Global dimension)

All 44 questions were translated into traditional Chinese by researchers at an educational institution in Taiwan. The Chinese version was used in this study with the developers' permission.

(Table 1) shows the frequency and percentage of the 84 participants' learning style preferences in each dimension. For example, three-fourths (63) of students were *sensing* learners, while only one quarter (21) of them were *intuitive* learners. Many more students (79) were identified as *visual* learners than as *verbal* learners (only 5, or 6%); in contrast, there were approximately as many *sequential* learners as *global* learners (44 vs. 40).

Table 1: Learning style preference distribution in this study

Learning Style Dimensions	**# Students (Percentage)**
Active/Reflective	38 (45%) / 46 (55%)
Sensing/Intuitive	63 (75%) / 21 (25%)
Visual/Verbal	79 (94%) / 5 (6%)
Sequential/Global	44 (52%) / 40 (48%)

● The Written Tests

The written portion of the midterm exam covered the programming constructs taught during the first eight weeks. Those constructs taught during the latter half of the semester were covered in the final exam. (Tab. 2) gives a list of specific programming concepts/constructs tested in the written part of the midterm and the final exams. To ensure content validity of the tests, all test problems were reviewed by an experienced Java programming instructor. The written part of both the midterm and the final tests contained 12 fill-in-the-blank questions and 5 short-answer questions. (Fig. 1) shows one of the problems given on the midterm test.

● The Hands-on Programming Test

Students were asked to write a complete Java program in the hands-on programming portion of the midterm and the final exam. On the midterm exam they implemented a Circle class which included radius as its private data member and five member functions, including the constructor and four methods used to set the radius, get the radius, and compute the area or the circumference of the circle. Students wrote another class called CircleTest to verify the functionality of the Circle class. In the program written by each student on the final exam, they were required to use overloaded methods and static methods in the class they defined to perform conversions between different length units. Again, specifications provided to the students as a basis for implementation were reviewed by the same Java expert mentioned previously to ensure correctness and completeness.

For each of the hands-on programming test problems, the instructor assigned partial credit to each of the several functions implemented by students. For example, if a student was able to define the *radius* data member correctly, s/he would get 5% of the total score; s/he would get another 15% if his/her program could compute the area of a circle correctly. In other words, the instructor examined the code written by a student carefully to identify any part of the program that was written correctly and assign partial credit accordingly even if the program did not produce correct output as a whole.

● The Debugging Test

Hands-on debugging tests were devised based on the findings reported in previous studies which identified common errors made by novice Java programmers (e.g., [1, 4, 9]). A complete program with a number of seeded errors was given for each debugging test.

The program used in the midterm debugging test, as shown in (Fig. 2), was intended to output the palindromes between 100 and 999 (e.g., 151, 202, and 848), but it contained four errors. First, the variable PalindromeCnt had not been initialized before it was used; second, the conditional expression of the if statement in the main method misused the assignment operator (=) where the double equal sign (==) should have been used for comparison of equality; third, each of the middle three methods of the Palindrome class returned a value to the calling object, but all three methods specified void as the return type; lastly, when a Palindrome object was instantiated in the main method, the parameter was missing. Students earned 25 points for successfully correcting each of the four errors.

The five bugs seeded in the program for students to debug on the final exam belonged to three other types of programming errors, namely, array subscript out of range (two seeded bugs), statements that should be inside the loop body was misplaced outside the loop (two seeded bugs), and an off-by-one error when using the mod (%) operator (one seeded bug). Correction of each bug accounted for 20% of the total score.

Table 2: Programming concepts/constructs tested in the written portion of the midterm and the final exams.

Midterm Exam	Final Exam
1. Arithmetic and relational operators (e.g., %, ==, !=, /=, ++)	1. Scoping rules
2. Naming rule for Java program files	2. Invocation of overloaded methods
3. Constructors vs. general methods	3. Parameter passing (pass-by-value vs. pass-by-reference)
4. Classes vs. objects	4. Creation of methods with unspecified number of arguments
5. How "empty statements" affect the execution of repetition statements	5. Static methods vs. non-static methods
6. Formatted output	6. Data hiding
7. Execution of a for statement	7. Memory allocation for class variables and instance variables
8. How a continue statement alters the execution of a loop	
9. How a break statement alters the execution of a loop	
10. Creating objects and invoking an object's methods	

The output of the following program is shown to the right. What should be filled in the blanks marked with ① and ②? (Hint: Use \t to separate the items that will be printed on the same line.)

```
N    10*N   100*N   1000*N
1    10     100     1000
2    20     200     2000
3    30     300     3000
4    40     400     4000
5    50     500     5000
```

```
public class formatOut {
    public static void main(String[] args) {
        int i;
        System.out.printf("____①____", "N","10*N","100*N","1000*N");
        for (i=1;i<=5;i++){
            System.out.printf("____②____",i,10*i,100*i,1000*i);
        }
    }
}
```

Figure 1: A fill-in-the-blank question in the midterm written test

```
public class PalindromeTest {
    public static void main(String[] args) {
        int count;                    //loop counter
        Palindrome ifPalindrome;
        int PalindromeCnt;            //累計 Palindrome Number 的個數
        System.out.printf("\n=====================================\n");
        System.out.printf("100~999 之間的 Palindromic Numbers：\n");
        System.out.printf("=====================================\n");
        while (count <= 999){
            ifPalindrome=new Palindrome();
            if (ifPalindrome.getHundreds()=ifPalindrome.getUnits()){
                System.out.printf("%d",    count);
                PalindromeCnt++;
                if (PalindromeCnt %5 ==0)      System.out.println();
                else    System.out.printf("\t");
            }
            count++;
        }
        System.out.printf("\n100~999 間的 Palindromic Numbers 共%d 個 \n",PalindromeCnt);
    }
}
public class Palindrome {
    private int hundreds, tens, units;
    public Palindrome(int threeDigNum ){
        hundreds=threeDigNum/100;     tens=threeDigNum%100/10;     units=threeDigNum%10;
    }
    public void getHundreds(){
        return hundreds;
    }
    public void getTens(){
        return tens;
    }
    public void getUnits(){
        return units;
    }
    public void display(){
        System.out.printf("%d,%d,%d\n",hundreds,tens,units);
    }
}
```

Figure 2: Hands-on debugging problem for the midterm exam

4. Results

4.1 Correlations between Test Scores and Learning Styles

A student's learning preference, as determined by the ILS questionnaire, was represented as a score between -11 and 11. A positive score in the active/reflective dimension means that the learner tended to be more active; conversely, a negative score indicated the learner's reflective tendency. The scores could be interpreted similarly in the other three dimensions. It should also be noted that the more extreme a number was, the stronger the learner leaned toward a certain tendency. For example, a score of -11 or -9 in the sensing/intuitive dimension represented a "strong" intuitive tendency; a score of -7 or -5 indicated a "moderate" intuitive tendency; whereas a score of 3 or 1 signaled a "mild" sensing tendency. According to ILS's scoring rule, a score could only be an odd number.

Students' ILS scores and test scores were analyzed with the SPSS statistical package (Version 17.0, SPSS Inc., Chicago, IL) to determine if there are correlations between students' learning preferences and their achievement in the exams. Specifically, each student had four ILS scores, each indicating a student's learning preference in a dimension. Besides, each student had three test scores, representing his/her performance on the written test, programming test and the debugging test. The two scores that a student obtained in the written part of the midterm and the final exams were averaged to derive his/her written test score. A student's programming score and debugging score were computed similarly. Therefore, there were four independent variables (i.e., learning preferences) and three dependent variables (i.e., test scores) involved in calculating the Pearson correlation coefficients and their statistical significance.

(Tab. 3) shows the results of the statistical analysis. As can be seen from the table, the only significant correlation found was that between the sensing/intuitive learning style and the written test score ($r = 0.236$, $p = 0.03$), indicating that *sensing* learners outperformed *intuitive* learners in the written tests.

Table 3: Correlations between ILS scales and test scores

Test / Dimension	Written Test	Programming Test	Debugging Test
Active/ Reflective	r = -0.15 (0.173)	r = -0.089 (0.419)	r = -0.027 (0.81)
Sensing/ Intuitive	r = 0.236* (0.03)	r = 0.098 (0.373)	r = 0.088 (0.427)
Visual/ Verbal	r = -0.016 (0.883)	r = -0.118 (0.286)	r = -0.041 (0.710)
Sequential/ Global	r = 0.159 (0.148)	r = 0.122 (0.270)	r = 0.108 (0.326)

4.2 Effects of Different Combinations of ILS Dimensions on Student Performance

A further analysis conducted in this study was to look at specific subgroups of students as determined by the four learning style dimensions. Only six subgroups were chosen for analysis. The remaining 10 subgroups were omitted because the size of each was too small for meaningful analyses. Each of the six subgroups was compared against the entire group. The mean score and standard deviation (enclosed in parentheses) of each subgroup on each test are summarized in (Tab. 4).

Table 4: Test scores of different learning style subgroups

Group	*n*	Written test	Hands-on Programming test	Debugging test
Overall	84	57.11 (22.75)	56.44 (31.40)	57.21 (32.01)
Reflective/ Intuitive/ Visual/ Global	6	47.00 (20.83)	42.33 (35.02)	34.50 (32.32)
Reflective/ Intuitive/ Visual/ Sequential	6	43.83 (26.99)	38.00 (34.91)	50.83 (29.92)
Reflective/ Sensing/ Visual/ Global	13	55.77 (29.62)	54.77 (34.43)	51.54 (36.69)
Reflective/ Sensing/ Visual/ Sequential	19	69.05 (21.55)	66.89 (34.41)	67.16 (32.72)
Active/ Sensing/ Visual/ Global	13	57.38 (21.45)	51.77 (29.45)	56.15 (34.21)
Active/ Sensing/ Visual/ Sequential	16	56.06 (18.10)	58.25 (25.05)	58.25 (27.60)

As shown in (Tab.4), the scores obtained by half of the six subgroups, namely reflective/sensing/visual/ global, active/sensing/visual/global, and active/sensing/visual/ sequential groups, were close to the overall score, whereas the other three subgroups scored much higher or lower. In particular, the reflective/sensing/visual/sequential group, which happened to be the largest subgroup, scored highest on each test among all the subgroups and was the only subgroup that scored higher than overall on each test. The two subgroups that scored lower than overall were the reflective/intuitive/visual/global subgroup and the reflective/intuitive/visual/sequential subgroup. When we excluded the visual/verbal dimension from subsequent analyses, it was found that students with the reflective/sensing/sequential learning style did better on each of the three tests. This finding more or less confirmed what had been revealed in previous studies (e.g., [2, 3, 12, 13]) that reflective learners tended to perform significantly better than active ones in introductory computer science or programming courses.

5. Conclusion

In this study we conducted an experiment to explore the relationship between learning styles, based on Felder-Silverman learning style model, and student performance in introductory Java programming courses. Though it yielded limited statistically significant results, it should help to provide more insight into how learning styles may affect student performance in programming and debugging. Moreover, unlike most studies that only examined correlations between student performance and single learning style dimensions, this study has attempted to investigate how different combinations of the four learning style dimensions might affect learning outcomes. Considering that learning styles can be an important factor in determining students' success in a computer science program, more studies are called for to broaden and deepen our understanding of learning styles and student performance in programming courses, as well as in other computer science courses in general.

References

[1] Ahmadzadeh, M., Elliman, D., & Higgins, C. "An analysis of patterns of debugging among novice computer science students," Proceedings of the10th annual SIGCSE conference on innovation and technology in computer science education, 2005, pp. 84-88. New York: ACM.

[2] Allert J. "Learning style and factors contributing to success in an introductory computer science course," Proceedings of the IEEE International Conference on Advanced Learning Technologies, 2004, pp. 385-389. USA: IEEE.

[3] Chamillard, A. T., & Karolick, D. "Using learning style data in an introductory computer science course," Proceedings of the thirtieth SIGCSE technical symposium on computer science education, 1999, pp. 291-295. New York: ACM.

[4] Hristova, M., Misra, A., Rutter, M., & Mercuri, R. "Identifying and correcting Java programming errors for introductory computer science students," Proceedings of the 34th SIGCSE technical symposium on computer science education, 2003, pp. 153-156. New York: ACM.

[5] Felder, R.M. "Reaching the Second Tier: Learning and Teaching Styles in College Science Education," Journal of College Science Teaching, 23(5), 286-290, 1993.

[6] Felder, R.M., & Silverman, L.K. "Learning and Teaching Styles in Engineering Education," Engineering Education, 78(7), 674–681, 1988.

[7] Fitzgerald, S., Lewandowski, G., McCauley, R., Murphy, L., Simon, B., Thomas, L., & Zander, C. "Debugging: finding, fixing, and flailing, a multi-institutional study of novice debuggers," Computer Science Education, 18(2), 93-116, 2008.

[8] Pillay, N., & Jugoo, V. R. "An investigation into student characteristics affecting novice programming performance," *SIGCSE Bulletins,* 37(4), 107-110, Dec., 2005.

[9] Robins, A., Haden, P., & Garner, S. "Problem distributions in a CS1 course," ACE '06: Proceedings of the 8th Austalian conference on Computing education - Volume 52, 2006, pp. 165-173. Hobart, Australia: Australian Computer Society, Inc.

[10] Soloman, B. A., & Felder, R. M. "Index of learning styles (ILS)," 1991 [Online]. Available: http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html.

[11] Soloman, B. A., & Felder, R. M. "Index of Learning Styles Questionnaire," 1996 [Online]. Available: from http://www.engr.ncsu.edu/learningstyles/ilsweb.html.

[12] T homas, L., Ratcliffe, M., Woodbury, J., & Jarman, E. "Learning styles and performance in the introductory programming sequence," Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, 2002, (pp. 33-37). New York: ACM.

[13] Zualkernan, I. A., Allert, J. & Qadah, G. "Learning styles of computer programming students: A middle-eastern and American comparison," IEEE Transactions on Education. 49(4), 443-450, 2006.

Acknowledgments

This research has been funded by the National Science Council of Taiwan, the Republic of China, under the grant numbers NSC 97-2511-S-003-014-MY3. The authors also wish to thank Tamkang University (Taipei, Taiwan) for allowing us to use the traditional Chinese version of the ILS questionnaire which they translated from English with the permission of its two authors. We are also grateful to Professor Hon-Ren Lin, Director of the Department of Information Management at National Taipei College of Business for reviewing test problems used in this study.

On the Use of Scrum in Project Driven Higher Education

Mia Persson[1], Ivan Kruzela[1], Kristina Allder[1], Olof Johansson[2], and Per Johansson[1]
[1]Department of Computer Science, Malmö University, 205 06 Malmö, Sweden
[2]Department of Psychology, Lund University, 221 00 Lund, Sweden

Abstract— *The use of scrum development methodology to promote learning in project driven, interdisciplinary higher education courses in computer science is examined by using theoretical notations and recent research results from the literature. Moreover, empirical data will be gathered from a case study on a research focused, project driven undergraduate course in computer science. Our contribution is to initiate a discussion aiming at increasing the pedagogical awareness on the use of the scrum methodology to promote learning in higher education. Moreover, a further refined scrum method, adapted and more fine-tuned to suit the particular needs in the learning situation at university courses in technology, is proposed.*

Keywords: education, software development, scrum, team projects

1. Introduction

The scrum development methodology has received significant attention recently (see e.g. [6], [13], [11], [16]). The aforementioned methodology has become one of the most popular and successfully applied software development methodologies nowadays in industry (see e.g. [13]). The scrum methodology belongs, together with methodologies such as extreme programming, to the set of agile software development methodology emerging in the early nineties as a reaction on earlier more heavyweight methods. At the heart of the scrum method lies a continuous communication between the developers and stakeholders due to the key feature of rapid prototyping in this method. Another important scrum feature is a highly motivated development team, which has a significant responsibility and huge impact during the development process.

However, as mentioned in [6], [10], agile methods have been applied in various situations, e.g., in post graduate education. In particular, the scrum method has recently been elaborated on in higher education course design in academic game development courses [11]. In this study we will focus on how the scrum methodology facilitates learning in research focused, project driven undergraduate courses in computer science. Project driven courses are well established in the curriculum for undergraduate studies in technology at most universities.

Moreover, recently there has been an encourage to engage students in current research project [3] aiming to ensure learning outcomes on the higher levels of the SOLO taxonomy, i.e., the relational and extended abstract levels. Recent pedagogical research also highlights the importance of that the students should be given the opportunity to work like the professionals within the research areas [7].

Several studies have been conducted aiming at investigating how to design appropriate learning activities in order to set up a learning environment which will work as a catalyst for learning among a group of students working together (see e.g. [4], [5]). Biggs [1] introduced the concept constructive alignment and this refers to a course where there is maximum consistency among the objectives, the teaching and learning activities and the assessment. To the best of our knowledge, there are no earlier studies considering the design of appropriate learning activities by using the scrum methodology in a research focused, project driven undergraduate course in computer science. The aim of this study is to contribute to an increased pedagogical awareness on how to use scrum in aligned undergraduate courses in technology involving ongoing interdisciplinary research projects.

2. Theoretical background

According to Marton et al. [8] developing a learner's capability of handling novel situations in powerful ways is one of the most important educational aims. As an illustrating example of this, consider the following example by Marton et al. [8]. Someone is standing in a lake with the water up to his knee and aiming at a fish in the water with a harpoon. He might aim at the fish where it appears to be, that is, where he actually sees it, or at a slightly adjusted angle, that is, where he thinks it should actually be if he takes the refraction of the light into consideration. As discussed in [8], the two different ways of acting in the aforementioned example, with one of them considered to be more powerful than the other, are actually based on two different ways of understanding the situation. In [8], it is further argued that pedagogical acts should always take as their point of departure the capabilities they are supposed to contribute to developing.

In order to achieve the aforementioned educational aim, Shuell [12] provides us with a good hint about our role as teachers when he discusses that if students are to learn desired (learning) outcomes in a reasonably good manner, then the teacher's fundamental task is to get students to engage in learning activities that are likely to result in their achieving those outcomes. Another statement, which is closely related to the aforementioned, is that it is helpful to remember that what the student does is actually more important in determining what is learned than what the teacher does [12].

By analyzing the aforementioned statements, we see that carefully designed learning activities are crucial in the learning situation of developing the student's capability of handling novel situations in powerful ways, and hence, this must be something that teachers should pay close attention to when developing courses. Specifically, how can I design a course where my role as teacher is more like a catalyst for learning?

Biggs [1] also argues on the importance of the learning activities in the course design when he introduces the concept of aligned courses, which supports the aforementioned focus on the learning environment set up by the teacher during a course. According to Biggs [1], in an aligned course, the teacher starts by specifying the intended learning outcomes in terms both of the topic content and also the level of understanding of the content (i.e., by considering the SOLO taxonomy) of the students. The next step is then for the teacher to set up an environment that maximizes the likelihood that students will engage in the activities designed to achieved the intended learning outcomes. The final step is then to choose assessment tasks that test how well individual students have attained the aforementioned outcomes, and then we arrive at the final grades.

As mentioned earlier, we are interested in finding appropriate learning activities in the particular case of a research focused project driven undergraduate course in computer science. According to Biggs [1], learning that takes place outside the class, with interactive group work, peer teaching, independent learning and work-based learning all contain relevant learning activities. We now seek to find a, possibly modified and fine-tuned, software development methodology compelling the aforementioned learning activities.

When working with software projects there exist conceptually two families of process models. The first consists of more formal and rigid models such as the waterfall model, the second is the family of agile software methodologies. The next section will discuss the differences of these two.

2.1 Software Development Methodologies

Software processes often stumble upon the complexity of designing, implementing and maintaining software systems, which match the requirements of stakeholders. Some of the complexity in the software process rises from "wicked"-problems. These problems can be described as problems which are difficult to define until they are nearly solved [13], [16]. Yeh [16] exemplifies "wicked"-problems in the context of software processes:

- Requirements which need an implementation before it can be fully derived and understood.
- There are no rules to determine when a solution is complete.
- Symptoms and causes cannot be distinguished.
- The problems are unique and may not be revealed before enough knowledge is gained from the system's domain.
- Definite problem definitions are not generally possible; some knowledge of the requirements of a system may be distributed across many stakeholders which make it difficult to collect the correct information needed to successfully implement a solution to the problem.

Traditional software process models such as the waterfall mode (see e.g [13] for a comprehensive survey of the waterfall model) often fails to accommodate the nature of "wicked"-problem due to the rigid way of first trying to define the problems with requirement analysis phases and then phases of implementing solutions for the problems. These shortcomings are now realized and new ways of thinking about the software process are established in the form of new evolutionary and agile methods [16]. This is where the scrum project management model fits in. The word scrum is not an acronym but a mechanism in rugby to get an off-ball into play again [9]. Scrum belongs in the family of agile software development methods and is based on an iterative incremental process [6], [9]. The output of each iteration is an increment of the whole product. Driving this iteration is a list of requirements called the product backlog. The core of scrum lies in its iterations. The team, working on a software project, takes a look on the requirements, consider the available technology and evaluate its own skills. It then collectively determines how to build the functionality, modifying its approach daily as it encounters new complexities, difficulties and surprises [9]. Each iteration, which with scrum terminology is called sprints, is initiated with a sprint meeting is hold with the stakeholders, product owner, scrum master(which can be seen as the project manager) and the team. The team can then choose which items of the requirements or product backlog they want to work with based in the evaluation of skills and the importance of the item [9]. After the sprint is initiated and the team has started working, daily meetings are hold which cannot exceed the time of 15 minutes. These meetings ensure the course of action and help the team to have an updated view of the project [9]. This attentive way of working by evaluating the process day by day is one of the things which make the method successful. Also, the close interaction with the stakeholders helps understanding the requirements of the product being developed. The way of working in iterative and incremental steps is also a good way of meeting the difficulties and complexities of software processes and it also makes it easier to deal with "wicked"-problems [9], [16].

3. Our case study

3.1 Project background

At the university under consideration in this study, project driven undergraduate courses in technology are a common and important part of the curriculum. This follows partly from the fact that there is an encouragement in increasing the employability of the students in higher education (e.g. due to the Bologna process). Moreover, there is a current ongoing discussion on how to involve students in current research projects, aiming at guaranteeing learning outcomes of high quality in higher education [3]. By learning outcomes of high quality we refer to a well established measure; a learning outcome of high quality can be categorized as belonging to the SOLO taxonomy's relational and extended abstract levels (see e.g. [2]).

In our project driven undergraduate course in computer science, the students took part in a newly initiated interdisciplinary research project, a joint work with the local hospital. More specifically, the hospital would like us to develop a new internet-based system for treatments of patients with anxiety and depressions. There are many potential advantages with such a system, some of them are an improved satisfactory level among the users of the systems and a more efficient health care section just to mention a few of them. At the moment, the contact by a doctor and the patient is by regular emails and physical meetings.

The teaching staff and a psychiatrist, our stakeholder, conducted the initial feasibility study. The teaching staff consisted of two assistant professors and one lecturer. One of the assistant professors was the initiator of the project and the other taught the courses in which the project was considered to be implemented. The lecturer participated as a support for the web developing part of the project. This part of the project consisted of a series of meetings during late January 2010 and the first part of February 2010.

3.2 Implementation of the Melencolia I project

The initial requirements from our stakeholder, a psychiatrist working at the hospital, were delivered at the beginning of our project at an initial meeting with the student group and the researchers and teachers running the project between the hospital and the university under consideration. During this meeting, also the scrum software development methodology was presented by the course responsible who has a background also working as an assistant professor in several undergraduate courses in software engineering courses including scrum in their curriculum. Scrum was presented as the methodology during the rest of this project and for some of the students scrum was already wellknown since it is often used in industry in software development projects. The desired system was identified as an information system using a web application for its user interface. The rest of the course was then organized as follows.

In early February 2010 the students were brought into the project. Scrum teams where made up which consisted of three third-year students and seven second-year students. The three third-year students formed one team (Team A) and the three-second year students formed two teams (Team B and C). The students had also other course activities in parallel with this project so the working time had to fit into their schedules. The teaching lecturer took on the role as scrum master and the psychiatrist the role of product owner. The students were faced with the entire problem of implementing an information system. They had no given infrastructure to work from. So the first part of the tasks was to assess the means with which they thought they could implement the desired system. The second year students had no prior experience of web development inside their curriculum. So they had to solve the technical problems as well as their own lack of knowledge in some parts. Literature and online resources were used to learn the skills needed as the need were identified. Students with more prior knowledge of some aspect also held seminars for the other students. This approach was mainly used to teach how the selected platform worked. The students showed own initiatives to learning outside that which was expected. An example of this is that they on their own initiative adapted part of the Melencolia painting as a logotype for the information system. The student initiated the project with their own feasibility study of what means they could use to solve the technical infrastructure. After some consideration of different techniques and platforms the following were chosen.

- Web server: Ubuntu [http://www.ubuntu.com/]
- Generation of interface: Java Servlet and JavaServer pages using Apache Tomcat [http://tomcat.apache.org/]
- Information storage: A relational database using MySQL [http://www.mysql.com/]
- Configuration management: Subversion [http://subversion.apache.org/]

The following months from February to May were divided into two sprints. Some allowances were made for holidays and larger exams. The whole project group met each dividing point between sprints. At these meetings the previous sprint was discussed, the results of the project so far demonstrated by the students and the priorities for the next sprint were revised. Between the project group meetings the scrum master held meetings with the teams at least once a week. The teams held their own meetings at an average of two meetings a week additional to the project group meetings and meetings with the scrum master. The contact between the teams and the product owner consisted mainly of the project meetings and online communication by e-mail and the project forums. When the teams felt the need for more intense communication additional meetings were held between the product owner ant the team. Information were shared by using not only by written text mimicking dialog but also diagrams, such as use case diagrams and real use case descriptions using UML and mock-ups of functionality by HTML and CSS. These artifacts were also used during meetings. Documentation of the system was made by using ER diagrams for the database design and class diagrams using UML to capture the structure of the program code. The priorities list was revised by discussion between mainly students and product owner. The teaching staff contributed only when they felt that some point might be overlooked and this point presented a risk of seriously delaying the project. The prioritized list consisted mainly of desired functionality. Additional to this were tasks that related to infrastructure demands and teaching. At a meeting some new functionality might be discovered and was then added to the list and prioritized. The product owner had most weight at prioritizing functionality. Students had a say in prioritizing so far as to say if some task had to be implemented before others due to infrastructure demands. Each sprint aimed at implementing some prioritized functionality that could be tested by the product owner or prototypes for further assessment of some functionality. Tables 1–3 demonstrate how the functionality was implemented of the different scrum teams during the project.

Table 1: Table showing the sprint work of team A. Note that * indicates a revision according to feedback from our stakeholder during sprint meeting.

Sprint	*Start*	*Team A*
1	7	Prototyping user interface
2	10	Construct treatments
3	13	Patient view of treatment
4	15	Patient view of treatment*
		Assign treatments
5	17	Construct treatments*
		Treatment status overview

Table 2: Table showing the sprint work of team B.

Sprint	*Start*	*Team B*
1	7	Infrastructure setup
2	10	User interface layout
3	13	Messaging interface
4	15	Formatted messages
5	17	Display of messages
		Message sorting
		Log in functionality

The final meeting with the stakeholder was held on June 6, 2010 and to this meeting the teams were supposed to complete the documentation of the system and also provide manuals for the future users of the system. There was also an open discussion on possible future work on the system.

4. Discussion

In this section we begin by analyzing our case study with respect to the learning activities that we found were naturally supported by using the scrum methodology in our project course. We then continue by propose a new more fine-tuned scrum methodology which suits the particular needs in a learning sitution in higher education project driven courses. We will discuss several features that we found important in order to gain optimum performance when using scrum in research focused learning situations.

Table 3: Table showing the sprint work of team C. Note that * indicates a revision according to feedback from our stakeholder during sprint meeting.

Sprint	*Start*	*Team C*
1	7	Design of modular system
2	10	Adapting user interface
3	13	Prototyping questionnaires
4	15	Questionnaire forms
5	17	Interface of questionnaire*
		Free form questions
		Log in functionality

We again recall that according to Biggs [1], learning that takes place outside the class, with interactive group work, peer teaching, independent learning and work-based learning all contain relevant learning activities for maximizing learning in learning situations in higher education.

We found out that by following the scrum methodology in our research focused project course, the aforementioned learning activities were naturally supported and implemented. Moreover, we found that one of the strengths with the scrum methodology is that it has some profound features which makes it an excellent choice as methodology promoting a deep approach towards learning among the students in the learning situation.

In our case study the teachers were not involved in the formation of scrum teams, the students themselves formed the teams. There were three teams and one of the teams constitutes the three third years students and the other two constitute the second year students in computer science.

4.1 Scrum sprints

We begin by consider the scrum sprints in our study. Recall that during a sprint, each team was supposed to develop a partially working subsystem and present their prototype of this subsystem to the stakeholder at a sprint meeting ending the sprint. Which subsystem to implement during a sprint was decided at the previous sprint meeting, i.e., each sprint meeting both ending a sprint and initiating the next sprint. During a sprint meeting the stakeholder and the students together agree upon and prioritize among the requirements to implement next and we found out that the teachers should ideally act more of an advisor and coach but also monitoring the prioritization so that a team does not end up in a situation were they have to wait for another groups implemented subsystem before they are able to continue with their implementation, we call this unwanted situation starvation. To prevent starvation is one of the most important tasks of the teachers when using scrum.

4.2 Scrum sprint meetings

We observed that there was a need for more frequent sprint meetings than scrum usually describes and one of the reasons for this was that the sprint meetings were found to be an excellent place for interactive group work among all the participants and for supervision of the students during their work with the system. We found out that for a group of 10 students two hours sprint meetings twice per month were appropriate. At these meetings, the student teams, the stakeholder and the teachers were gathering in a small room around a round table and the only equipment was a large white board and a video projector, which was used for the prototype presentations of the students. The students brought their own laptops. Each of the sprint meetings follows the same structure, namely that each team presented their developed prototype in turn (i.e. partial subsystem) and the other student teams, the stakeholder and the teachers were encouraged to reflect upon the work. The stakeholder was allowed to make some test cases and provided directly feedback on the prototype and we observed that this rapid prototyping and the quick feedback of the stakeholder, which actually lies at the heart of scrum, put the students communication with the stakeholder in focus and thus yielding a learning situation during the sprint meetings where the students were completely in focus, i.e., the key features of the scrum method naturally support a student-focus in the learning situation. It is well known that student-focused teaching is a necessary condition for students to adopt a deep approach towards learning [14], [15].

We also found that the students initially were excited when they heard that we were going to deploy a software development methodology which is well known and often used successfully in industry software development projects and this was certainly an experience they would put in their CVs. Moreover, they also found that during the project they became highly motivated by the committed stakeholder and by the research focus of the project. In other words, when the students found that the their project work was relevant not only for them but also for our stakeholder, the hospital and the patients, and for research community on internet based health care support system in general, they felt a great responsibility of delivering a high quality system obeying high usability, security, and etc. We also observed that independent learning was taking place in this project when the students felt that they were working with moving the research frontier forward and the teachers were more of an advisor for them in this process. Furthermore, during our second scrum meetings our stakeholder invited a researcher working with internet-based health care support systems at another nearby university and he provided helpful feedback on the students ideas on what type of system we were to develop and also on open problems and the current state of the art within this area.

4.3 Scrum daily meetings

We found out that the student teams formed informal meetings approximately twice per week. At these meetings, the students planned and coordinated the sprint work and also prepared the scrum meeting presentations. Moreover, we found out that at these meetings the more experienced students began to act like mini scrum masters and teached the other students, thus we found that peer teaching as well as interactive group work were naturally supported in scrum. As the course went by, we found that the more experienced students also worked with supervision across the initial scrum teams formation, i.e., the third year students supervised also the second year students during their informal scrum meetings. The supervision was both in the form of recommending relevant literature for conducting the sprint work but most important, they also took an active part in the important process of problem solving among the second-year students. The third year students acted like mentors in this higher-order cognitive process that required both the modulation and control of more routine or fundamental skills of the students.

5. Our proposed scrum

We posit a set of rules of thumb that we found out lead to a successful scrum project in higher education courses in computer science.

- It is recommended to use scrum on interdisciplinary research projects.
- There should be one separate stakeholder and one separate scrum master since it is most likely that this will increase the students motivation.
- The stakeholder must be committed to the project since this increases the student motivation.
- One of the scrum masters (we recommend one of the teachers here) must monitor and prevent starvation. This can be achieved by having shorter sprints. We recommend that there are at least two scrum sprint meetings per months.
- A product backlog should be kept during the whole project. During the sprint meeting, one of the scrum masters (in our study one of the assistant professors) updates the product backlog according to the feedback from the stakeholder when the prototypes are tested.
- The students should themselves form the scrum teams.
- It is recommended to have mixed classes in the project, i.e., students at different stages in their education and let the more experienced students act as "mini scrum masters" and supervise the other students.
- It is recommended to use scrum also on small teams (e.g. 10-15 persons).

- It is difficult to have daily scrum meetings in higher education courses due to students other courses, instead encourage the students to have approximately two informal scrum team meetings per week.
- We found it possible to omit the burn down chart.

6. Conclusions and future work

We highly recommend using scrum in a research focused, project driven undergraduate courses in computer science. However, from our case study we have gained some more insights on how to fine-tune the scrum methodology from industry in such a way that maximizing learning in learning situations in higher education. We found it suitable to implement the scrum methodology in external interdisciplinary project, i.e., it is a strength to have a separate committed researcher acting as a stakeholder and an assistant professor acting as a scrum master, and this, together with the motivated students, formed a creative learning environment. Scrum was found very suitable for also smaller academic projects but daily scrum meetings were not possible due to the students other courses. We recommend also shorter sprints in the academy since this is a natural way to implement more interaction between the students and the teachers when using scrum. Moreover, we also found that the phenomenon of starvation is more easy to handle for the teachers with shorter sprints in undergraduate courses.

We conclude that the stakeholder of our project was highly satisfied with the developed system and it actually became more advanced than expected from the beginning of the project. By this project, the university and the hospital under consideration now got a new system to further work on together in new future higher education courses. Moreover, the project also resulted in a company founded by some of the students participating in the project.

References

[1] Biggs, J. B. Teaching for quality learning at university. Open University Press/ Society for Research into Higher Education, 2003.
[2] Biggs, J. B. and Collis, K. F. Evaluating the quality of learning – the SOLO taxonomy. New York: Academic Press, 1982.
[3] Geschwind, L. Utbildningskvalitet och forskande lärare - hur ser sambandet ut? Arbetsrapport 2008:86, Lars Geschwind, Institutet för studier av utbildning och forskning, Stockholm, 2008.
[4] Karlström, D. Introducing Extreme Programming — An Experience Report. In Proc. 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP2002), 2002.
[5] Madeyski, L. Preliminary Analysis of the Effects of Pair Programming and Test-Driven Development on the External Code Quality. In Proc. 2005 Conference on Software Engineering: Evolution and Emerging Technologies, pp. 113-123, 2005.
[6] Mahnic, V. and Drnovscek, S. Agile Software Project Management with Scrum. University of Ljubljana, 2005.
[7] Marton, F., Hounsell, D. and Entwistle, N Hur vi lär. Raben & Sjögren, 1986.
[8] Marton, F. and Tsui, A. B. Classroom Discourse and the Space of Learning. Lawrence Erlbaum Associates, 2004.
[9] Schwaber, K. Agile Project Management With Scrum. Microsoft Press, Volume 1, 2004.
[10] Rico, D. F. and Sayani, H. H. Use of Agile Methods in Software Engineering Education. In Proc. AGILE Conference, pp. 174-179, 2009 Agile Conference, 2009.
[11] Schild, J., Walter, R. and Masuch, M. ABC-Sprints: adapting Scrum to academic game development courses. In Proc. of the Fifth International Conference on the Foundations of Digital Games (FDG 2010), pp. 187-194, 2010.
[12] Shuell, T. J. Cognetive conceptions of learning. Review of Educational Research, 56, pp. 411-436, 1986.
[13] Sommerville, I. Software Engineering. Addison Wesley, 2010.
[14] Trigwell, K., Prosser, M. and Taylor, P. Qualitative differences in approaches to teaching first year university science. Higher Education, 27, pp. 75-84, 1994.
[15] Trigwell, K., Prosser, M. and Waterhouse, F. Relations between teachers approaches to teaching and students approaches to learning. Higher Education, 37, pp. 57-70, 1999.
[16] Yeh, R. T. System development as a wicked problem. The impact of CASE technology on software processes, 3, pp. 15-29, 1994.

Computer Supported Collaborative Learning for helping novice students acquire self-regulated problem-solving skills in computer programming

S.R. Brito[1,2], A. S. Silva[1, 2], O. L. Tavares[3], E. L. Favero[4], and C. R. L. Francês[2]
[1]Cyberspace Institute, Federal University Rural of Amazônia, Belém, Pará, Brazil
[2]Laboratory of High Performance Networks Planning and Postgraduate Program in Electrical Engineering, Federal University of Pará, Belém, Pará, Brazil
[2]Informatics Department, Federal University of Espírito-Santo, Vitória, Espírito-Santo, Brazil
[4]Post-Graduate Program in Computer Science, Federal University of Pará, Belém, Pará, Brazil

Abstract - *This paper presents the integration of a viewer and simulator of programs into an environment learning to support learning programming in initial classes of Engineering, Computing and Information Systems Courses. The integration proposed was based on a redesign of the architectureto combine resources avaliable in learning environment with an automatic evaluator of programs and with a new resource that promotes collaborative feedback through peers review. The new architecture resulting from this study is based on principles of collaborative feedback as a way of developing self-assessment skills in the programming disciplines.*

Keywords: Learning computers programming, collaborative feedback, self-assessment skills, self-regulated learning, automatic evaluation of programs.

1 Introduction

The development of skills and abilities needed for the construction of computer programs requires a significant effort by students and teachers of computer courses. In initials activities of disciplines and projects that use computer programming, we expect at least that students will be able to construct solutions to solve simple daily problems. Despite all efforts, difficulties encountered in subsequent disciplines are still visible, which require computer programming skills.

In order to build these skills, teachers devote efforts in teaching of syntactic and semantic aspects of programming languages. Often, the didactic approach involves the presentation of sample programs with structures similar to those applied in programming exercises. According to Menezes et al. [1], this strategy is based on thet fact that students need to create, in his/her memories, structural patterns that can be used to solve different problems.

Some of the obstacles faced in teaching programming are described by Sajaniemi and Kuittinen [2] and are often associated with difficulties in understanding the language syntax and abstract concepts like loops, pointers, arrays and other formal constructions applied for the first time. In most cases, when student perceives that he/she is not evolving in discipline, he/she feels discouraged and his/her performance becomes increasingly smaller.

By articulating the assumptions of self-regulated learning, we find three main stages [3][4]: planning, implementation and self-reflection. These phases are also present in computer programming laboratories, promoting and contributing to intellectual autonomy of students and stopping teacher to impose his/her knowledge and his/her way of working to the student. When the student stops to receive ready and finished concepts, he/she learns to find answers and search for solutions with freedom and responsibility, i.e., working to develop his/her autonomy, regulating his/her goals, strategies, motivation, cognition and behavior.

In the process of self-regulated learning, several authors agree that feedback is considered as a key aspect [5][6][7][8][9]. For our research, the central focus is the importance of monitoring and feedback to encourage self-regulation in a computer supported learning environment for novice programming learners. Besides seeking to improve the quality of feedback messages, we propose to explore learning scenarios that involve self-assessment skills, emphasizing situations where the student provides feedback to the solutions of his/her peers (teachers, aides, students and other participants in the process teaching-learning). As a result, we present an architecture and a system with focus on learning programming initials skills and we show that students who develop self-assessment skills, through the exercise to provide feedback to their peers, achieve significant progress in learning, even when the teachers' feedback is poor, which happens often in classes with high number of students.

2 Challenges of learning computer programming

In the following subsections, we describe some problems that are relevant challenges for the context of this research.

2.1 A. The importance of reflection in programming labs

When based on experimentation and construction, the learning encourages the development of cognitive skills and attitudes of high intellectual level. Therefore, the construction of computer programs cannot be seen only as the application of knowledge for building and implementing a computing solution without extensive discussion about its results.

As well as in disciplines of physics or mathematics, the construction of a solution through an algorithm or a program written in a programming language is seen not only as the production of a set of results, but as a moment of work, reflection, analysis, inquiry, interpretation, exchange, decisions and conclusions, which are provisional or not. More than an activity that causes curiosity, the process of building a solution should be considered a pedagogical practice in which "doing" is important, but the reflection is essential. To Stamouli and Huggard [10], it is important to provide opportunities for decisions review because, although the understanding of program's logical correctness is significant, the analysis of the solution by student directly affects the construction of strategies to solve problems to be faced later.

2.2 Teacher's overload in mediation learning

Besides the evaluation of program correctness, teacher can ask student about his/her solution so that he/she is conducted to describe his/her program or steps taken for his/her construction, making student to reflect on it. It can also provide new activities from the solved problem or encourage peer review.

The assessment is important because it can stimulate students to reflect on their solutions and about the process adopted to solve the problems. Requiring students to keep a description of their activities is a way to encourage the reflection. On the other hand, mediation is difficult when the classes are large. In cases when syntheses or reviews explanation about their solutions are requested, there is an overload of work for teacher to assess and, therefore, a delay in the teacher's feedback to these descriptions. It happens mainly when the process of evaluation and feedback is focused on the teacher.

In student-centered learning, the feedback process can be designed to include peer review or it can be facilitated by computer technology which provides elements that can contribute significantly to the feedback be internalized for self-regulation. Moreover, the teacher's feedback should not have reduced importance in self-regulation process, which is justified for at least two arguments: self-assessment integrated with teacher's feedback contributed to increase identification and correction of mistakes by students [11], and teacher's interventions can increase motivation and cognition, resulting in development of self-regulation skills [12].

2.3 The feedback limited to the transmission of information

According to Nicol [8], a greater student's commitment and responsibility in teaching-learning process imply changes in the way of conducting the assessment and feedback process. In a student-centered learning approach, the feedback cannot be conceived as a process of transmission, in which teachers provide information in order to conduct the correction and improvement of academic work.

The conception of feedback as a process of "transmission of information", often centered on the teacher, has been contested by several researchers [13][14][15][8][9]. Some main arguments are:

(1) there are strong indications that the feedback messages are complex and difficult to decipher, therefore, students need opportunities (e.g. through discussion) to understand and use the feedback to adjust their performance [15][16];

(2) it is a way to ignore the its relationship with the student's motivation [8], once the feedback from peers, teachers and mentors influences on how students see themselves and how they learn or build their own strategies [17];

(3) it focuses on teacher effort to produce feedback, which increases the overload of teacher's work, according to the number of students and classes, not always ensuring its effectiveness;

(4) when it not leads to a reinterpretation by student about the causes of failures in their solutions, not contributing to student's perception about their own effectiveness [14].

3 Basic principles for the proposed architecture

In the following subsections, we describe a conceptual framework to design the proposed architecture.

3.1 The role of feedback in the process of self-regulation learning

Researchers argue that students with self-regulation learning skills are more persistent, inventive, convict and enterprising [18][19], and are less dependent on external support from teacher [18]. According to Nicol [8], development of self-regulation skills can be facilitated by learning environments.

Our concern is to provide conditions for the feedback to help in the development of autonomy. For this, the feedback should be interpreted, (re) constructed and internalized so that it can have a significant influence on student learning [16].

Thus, we consider the feedback as a way to provide the student the necessary mechanisms to clearly understand [15]: (1) which are the course objectives, i.e. what is considered as a good performance, a learning goal to be internalized and achieved, (2) how current performance is related to the desired performance (for this, students should be able to compare the achieved performance to the desired performance), (3) how to act to reduce or eliminate the gap between the current performance and the desired performance.

To reduce the gap between the educational and the desired objectives , Sadler [15] suggests that students should possess "some of the teacher's skills evaluating". For some researchers [13][14], this observation shows that, besides improving the quality of feedback messages, we must invest in effort for building self-assessment skills. These arguments explain the fact that students with self-assessment skills achieve greater progress.

3.2 The importance of observation and collaboration

In order to make learning more active, Shang et al. [20] suggest to expand the learning experiences, like the creation of small groups of students, suggesting questions or putting them in situations of decision making. As example of collaborative experiences, Lazakidou and Retalis [21] propose the usage of computer-supported collaborative learning strategies for helping students acquire self-regulated problem-solving skills in mathematics.

After solving a programming problem, students may be asked to describe the way chosen and the decisions taken to discuss with their peers. For example, in the methodology proposed by Menezes et al. [1], adapted from Polya [22], students write their own understanding of problem. The text produced can help to promote discussions in small groups. These discussions are precious even for those not involved in dialogue. This opportunity, when the experience is related to the dialogue, offers students a new perspective on their beliefs and values, helping them build new possible meanings for their experiences.

We consider four ways of learning strategies presented by Shang et al. [20]: reflection, dialogue, observation and doing. The combination of these learning modes adds value to the process, making it more attractive and significative for the students, hence the importance for replacing most the traditional lesson by learning scenarios involving experimentation, discovery, creating solutions, discuss and knowledge systematization.

3.3 Process of problem solving

To solve a problem, the student must have a clear idea of the characteristics of the problem, attributes and rules, i.e., is necessary that the student understands the process of problem solving as a transformable system.

Just as occurs in the teaching of sciences and mathematics, several models of problem solving have been proposed to support the teaching and learning programming. These models have in common the definition of steps or phases in process of solving problems. For example, from an adaptation of ideas from Polya [22], Menezes et al. [1] suggest the following steps in the specification of what they called Computer Supported Programming Learning: (1) Understanding the problem, (2) planning, (3) Development, (4) Evaluation of process and its results, (5) Socialization of the results.

According Lazakidou and Retalis [21], the common point of the models support the teaching of disciplines to solve problems is initial phase: when the student is oriented to identify objectives of the problem, it is also oriented adjust their actions, considering the goals identified. That is, the student builds the internal model of problem and this model should be confronted with the goals identified. The step of understanding the problem involves definition of input data and their properties, expected results and relationships between input data and expected results.

After building the internal model of the problem, and know some rules associated with the problem, the student performs the planning of the instruction set that must be applied in solving the problem (even mentally). With the planning of its solution, the student builds some instructions, such as reading and writing data. It is the process of coding and testing solutions in a programming language.

The assessment of instructions using a computer environment or mental form, should lead to selecting the most appropriate instructions to the objectives of problem. These instructions or block of statements are registered in memory as pattern positive and will be applied to solve the current or other problems. To recover these patterns of human memory, Lazakidou and Retalis [21] suggest that mediation during the process of solving problems can include meta-cognitive strategies, as support, which can be verbal or not.

Therefore, assessment is a process of reflection at every step of process builds programs in order to consolidate the strengths and weaknesses each step. Thus, during the review or peer review, students can also go back the instructions or commands, confronting results (even partially) with the objectives and actions. The diversity in understanding the rules of a student to another, also contribute promote interaction in a cooperative environment. This process can derive new actions and changes in the initial instructions of the student's solution. As result of operative action self-regulating, through a selective process, actions compatible with the goals are registered. Viewers and simulators algorithms and programs can contribute for student's perception about the instruction or block of statements that are more suitable for the solution the problem.

Thus, while builds instructions, the student develops an orderly and sequential reasoning especially suitable for learning algorithms.

The socialization the result is considered an important experience because it is an opportunity of peer reviews, where students send and receive feedback. Therefore, it is essential provide an explanation of the developed solution: the objectivity and clarity of ideas are tested and experienced.

3.4 Construction of self-efficacy beliefs

The self-efficacy beliefs refer to personal evaluation or perception regarding their own capabilities [23]. The relevance of self-efficacy beliefs about motivation and performance are common in studies on self-regulation of learning [24][25]. In this direction, we emphasize:

(1) self-efficacy's perception, i.e., observation of other students who reached good results suggests to the student that he can handle similar challenges, motivating him to solve the problems [23];

(2)appropriate degree of difficulty of the task, where the intent is to provide a sufficient variety of activities, preventing the student faced challenges with difficulty just too high that can causes bad effects in your motivation [23];

(3)progress perception can provide the student the belief that it can solve new problems - for this, must be pointed out the progress through an effective feedback;

(4)social comparison should be avoided [23], i.e., must be avoid supply the same tasks and demand the same pace in solving problems, form groups of students according to their capacity and excessively stimulate the competition in class - in general, the goal is promote mechanisms that contribute to the student's perception about their own performance;

(5)the teacher should be informed of situations where their intervention is needed, i.e. in situations where the programming environment does not provide sufficient evidence to support the student's progress, verbal persuasion [23] must be present in teacher feedback. Previous researchs [12] indicate that the intervention increases motivation and cognition, towards the development of self-regulation skills.

We recognize that self-efficacy beliefs alone are not responsible for the success of students, but we propose adopt this principles so that students make using more appropriate strategies, investing more effort in solving the problems despite obstacles and failures. The effort and perseverance combined with right strategies can achieve, in general, better performances [4] [23][24][25][26].

4 Proposed environment

This research started with the development of a simulation and visualization programs called JavaTool [27], in order to use programs animation to support the learning of algorithms and programming in initial disciplines of Engineering and Computer Science. The integration of the Moodle [28] with JavaTool was justified by integration of resources and implementation of new features that allowed the JavaTool's invocation from Moodle [28][29]: examples, tasks and quizzes programming. We incorporate the method proposed by Moreira and Favero [30] towards assessment automated of programs. The resulting platform this work was used in four classes of algorithms and programming in courses of Computer Science and Information Systems.

Based on the theoretical principles of self-regulated learning, we found that most efforts to provide feedback was centered teacher and learning environment, i.e., the feedback so far has been conceived as a process of transmitting information to guide the improvement and correction of student solutions. Since then, requirements were identified towards the construction of a new architecture for the environment proposed by Mota et al. [27].

4.1 E-learning environment architecture

The architecture was built to accommodate characteristics of tutoring (through examples and simulations these examples), simulation (of solutions students), student tracking and support to collaboration.

The integration of Moodle [28] with JavaTool allows access resources and activities. The tutoring subsystem uses a resource type called programming example and others subsystems (Simulation, Automatic Evaluation and Collaborative Feedback) uses new types of activities incorporated into Moodle (Fig.2). These subsystems are described below.

For teachers, the architecture provides a subsystem to support the planning and monitoring of students. Addition to facilitate teacher's planning, the information given by the teacher in planning must serve as inputs to the perception subsystem of learning objectives used by student.

The simulator allows the student to visualize graphically and textually, through the explanation of steps, the execution of each example available in your learning space, beyond their own solutions. Through the actions of the student and their solutions, the Subsystem Monitoring and Evaluation provides automatic feedback with additional information that might contribute to learning self-regulation.

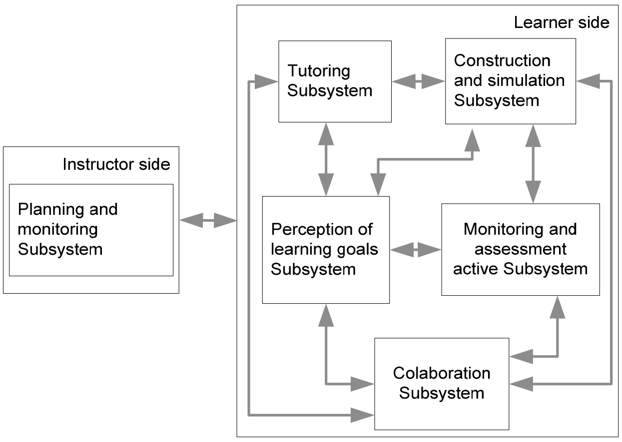


Fig. 2. environment architecture of programming learning with support for cooperative evaluation and feedback.

Finally, the subsystem supports the collaborative interactions between participants, so that feedback is not restricted to teacher or to learning environment. Thus, students do the exercise to give and receive feedback.

4.2 Tutoring Subsystem

Along a programming course, is important that students have access to basic learning materials and content and examples. Specifically to disciplines of programming, we implemented a new resource type in Moodle [28], called "programming example". In using this resource, the teacher registers examples of programs or save a solution implemented by the student on the basis of examples. Examples are available as resources and can are used according to teacher criteria or free exploration by students. All examples can be visualized with the code animation.

Were also developed two new types of activities: programming task and programming quiz. The programming task is a problem whose solution will be implemented in to simulation environment. The quiz contains several questions, made or selected by teacher, and the solutions for questions are also implemented in to simulation environment. The student can be consult the questions and the history of attempt with their assessment (Fig. 5), which may have been generated by automated evaluator or by teacher.

Pergunta	Texto da pergunta	Histórico das respostas	Melhor nota	
1	Descubra o maior de três números.	#1(8), #2(7),	8	Responder
2	faça o fatorial de n.	#1(4),	4	Responder
3	Imprima um numero aleatorio entre 1 e 10.	#1(3), #2(5), #3(9),	9	Responder

Fig. 5. Initial view the programming quiz (screenshot of student's screen)

The teacher can relate programming tasks and programming quiz with examples of programs so that students can use the same structures of the examples in their solutions. E.g., if along simulation the student perceives a syntax error in repetition structure their solution, it can search a example that uses the same structure are trying to use in examples base.

4.3 Simulation and visualization programs

A study based on the functioning of the cerebral hemispheres confirms the hypothesis is that visualization algorithms can help to better understanding [31]. According this study, the left hemisphere processes verbal and logical information while the right hemisphere processes visual and spatial information. Therefore, given the source code, read and view a graphical representation this code can stimulate both hemispheres, increasing the chance of better code understanding.

In that direction, the programs and visualization systems of programs and algorithms appeared, whose benefits justify the researches broadcasting in that area [32]. For other hand, in one of the most important studies about the effectiveness of those systems [33] concluded that the way students use the visualizations is more important than the animations and images in itself. The most recent study of evaluation of those systems realized by Urquiza-Fuentes and Velázquez-Iturbide [34] uses the taxonomy proposed by Price et al. [35]: Algorithms Visualization (AV), to encourage or present statically the algorithm; Programs Visualization (PV), to encourage or to present statically the algorithm or data structures; Systems based on Scripts (SBS), where the users select the code passages that they want to visualize; Interface Systems (IS), it doesn’t generate visualization, they just interact with the user (in some cases invoking SBS), and; the Compilers Based Systems, that insert the visualization actions automatically, simplifying its use, but limiting the user’s actions.

In class of viewers programs, the simulation in Java Tool [29] occurs through the animation of source code and can be executed from examples, tasks or programming quiz. In Figure 6, we see a simulator screenshot (solving programming question).

The region "A" contains simulator toolbar, where they are located animation menu and history, help, information and save buttons. The editor is located in region 'B', where the student can see keywords of the language in highlight (subset of Java language). The region 'C' contains the simulator's control panel with options to control the animation's execution, to save program, get help and information. The region 'D' contains the outputs of program run. Finally, 'E' and 'F' (disabled during the simulation) contain the spaces responsible for graphical and textual representation of the instructions performed.

The simulator allows textual and graphics animation, towards the student can understand the logic with an overview of program. Among Java's resources available programming environment are primitive types, arrays, selection and repletion control structures, some methods of class Math's and the creation and simplified methods invocation.

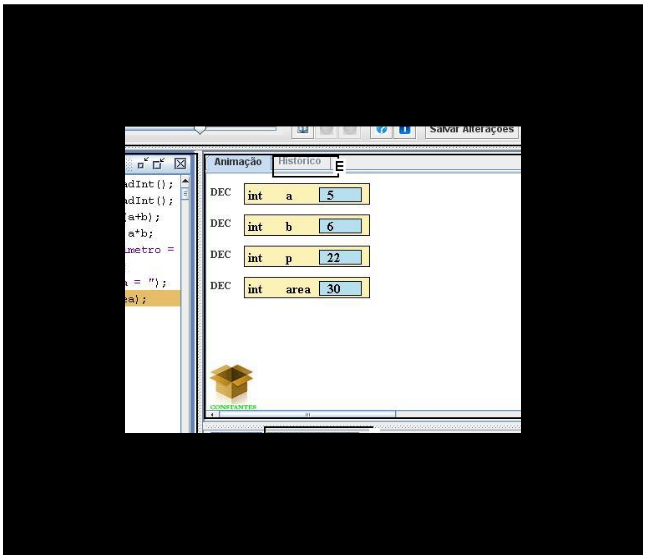


Fig. 6. Preview JavaTool in Moodle (Screnshot student's screen).

4.4 Monitoring and Assessment Subsystem

The evaluation subsystem allows automatic or manual assessment, with posting of notes and teachers feedback. The model of automatic evaluation [30] combines the use of a assessment of the code's complexity by statistical technique of multiple linear regression, with complexity indicators, with a code tester by input/output. The simulator is responsible for activating the automatic evaluation.

The indicators used by code evaluator are software engineering metrics. The technique consists in extracting from student's solution an value obtained by applying each metrics described by Moreira and Favero [30]. The subsystem assessment submits these values to multiple linear regression model, which consists of a linear equation previously obtained by training on test base. In this case, the metrics are the formula entries which calculate directly the student's grade. Finally, is checked if the solution returns the expected result for previous entries informed.

For the correct operation of the evaluator, the teacher must register a solution considered "model answer" to the problem. Thus, if there is no compilation error, the system evaluates and obtains a value for the student's solution, which can be modified depending on assessment's outcomes make do tester of input/output. The final assessment for each question, with history of attempts, is displayed in student's interface. The subsystem uses the highest score obtained on each question towards calculate the final score of programming quiz.

Automatically, when a student's solution achieves the highest score it becomes part of the tests base. In addition, the teacher can also select the student's solutions that should be incorporated into the test base. The test base contains the solutions used by the evaluate subsystem and by tutoring subsystem that is responsible for presenting examples.

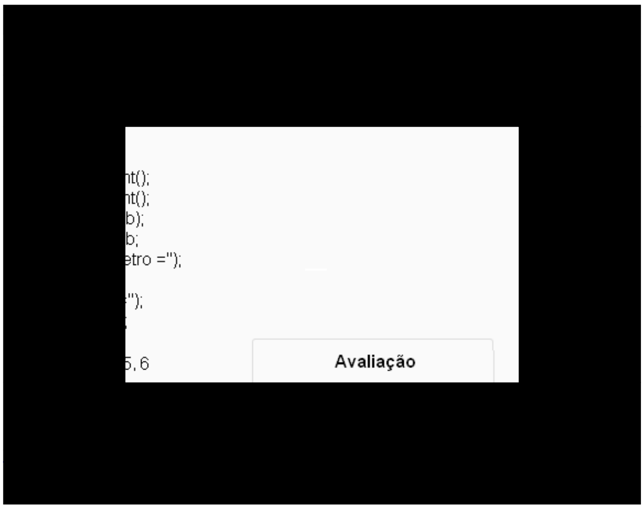


Fig. 7. Automatic assessment to student's solution (student's interface).

4.5 Subsystem collaborative assessment

The programming environment integrated into Moodle [28] allowed the use of interaction technologies that already exist on the platform.

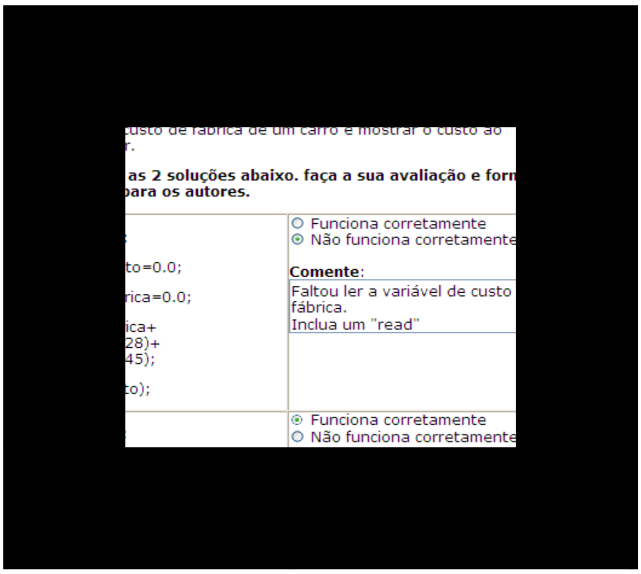


Fig. 8. Activity collaborative: peer feedback.

However, in order to exercise and develop skills of self-assessment, we implemented a new feature, called "collaborative evaluation", where students can provide feedback to the solutions of their peers based on their knowledge.

With the new functionality (Fig. 8), students can view feedback provided by their peers and contribute to the solutions of other students. The selection of solutions assessed by peers is done randomly from a database of solutions performed by students. Thus, the student immediately receive information about their assessment and it can consider relevant or not the feedback given.

4.6 Planning and monitoring subsystem

The subsystem planning and mediation is derived from the integration of new functionality implemented and incorporated to Moodle and to JavaTool, from the perspective of teacher. In addition to planning of contents and activities, the teacher uses this subsystem's resources to track the uses of resources and activities by students.

Especially in automated assessment, the teacher can view instantly the outcomes (Fig. 9), which allows to adopt different strategies in face the learning difficulties presented in its accompanying map, such as creating pairs or students groups during a class laboratory practice.

Fig. 9. Visualization of solutions (history of student attempts).

5 Conclusions

Two important aspects of assessment were considered so far: usability and educational effectiveness. According Kulyk et al. [36], usability's should be part of the design process and implementation. Therefore, there should only be performed when the learning environment is ready for use by students. Thus, we consider assessment phases: (1) the first prototypes were evaluated with heuristics inspections and the outcomes these evaluations were used to improve the system, (2) the version improved system was evaluated on consultation techniques, controlled experiments and observational studies and these results helped to correct bugs and improve the system, and finally, (3) observational studies were used to evaluate the use of the environment in real scenarios of learning in laboratory.

From the standpoint of usability, we consider: informal assessments, where students' opinions was captured after system use [29], heuristic evaluations performed by specialist teachers with the use of interactive resources, visualizations and simulations [27][29][30], observational studies where developers and evaluators observed the use of the system and reported important aspects of use of the system towards the specification of new requirements [29]; consultations through questionnaires to students.

The diversification of strategies for usability evaluation was important because only observational studies and questionnaires applications are realized in environments partly controlled by evaluators and therefore the information captured are more restricted. The diversification of strategies for usability evaluation was important because only observational studies and the questionnaires applications was realized in environments partly controlled by evaluators and therefore the information captured was more restricted.

In controlled experiments, the environment and student's tasks are controlled towards answers provide information on efficiency, ease of use and others usability indicators. However, the records of the actions of students are maintained even in uncontrolled experiments for comparison of performance and usability through the system logs.

As for educational effectiveness, Hundhausen et al. [33] show that the effort devoted by students to tasks related to visualization is more important than the visual content displayed by viewer and simulators. Naps et al. [37] developed a taxonomy for different levels to interact with these technologies. In this taxonomy, the authors suggest a hierarchical structure, where a high level of student involvement leads to greater educational benefits. Every level, from the visualization level, includes all previous levels.

After integration with Moodle, a new evaluation was performed using a taxonomy of Naps et al. [37]. In this evaluation, and in comparison with other environments, the new environment was rated at level 4, including all categories: View, Reply, Modify and Build. The level 5 can be reached from the adoption of a methodology where the student presents your solution to the class, posting your solution and commenting in forums or other mechanisms of interaction available.

The environment was used in four courses in algorithms and programming, with classes of 15, 29, 42 and 47 students. Questionnaires were used at the beginning of the course, aiming to capture the students' initial knowledge, development of logic and organizational skills. Structured questionnaires also were applied at end of course, with the aim of assessing the motivation for use of environment learning and technologies interface used..

6 References

[1] C. S. Menezes, O. L. Tavares, R. A. Nevado, D. Cury. “Computer Supported Co-operative Systems to support the problem solving - a case study of learning computer programming”. In: The 38th Annual Frontiers in Education (FIE) Conference, 2008, New York. Proceedings of The 38th Annual Frontiers in Education (FIE) Conference, 2008.v. 1.

[2] J. Sajaniemi and M. Kuittinen. “Program Animation Based On The Roles Of Variables”. Proceedings of The 2003 ACM Symposium on Software Visualization, 2003.

[3] B. J. Zimmerman. “Academic studying and the development of personal skill: a self-regulatory perspective”. Educational Psychologist, v. 33, n. 2/3, p. 73-86, 1998.

[4] D. H. Schunk. (1989). “Self-efficacy and cognitive skill learning”. In: AMES, Carol & AMES, Russell (eds.) Research on Motivation in Education. Goals and Cognititons. New York: Academic Press, Inc., v. 3, p. 13-44, 1989.

[5] D. L. Butler and P.H. Winne. “Feedback and self-regulated learning: a theoretical synthesis”, Review of Educational Research, 65(3), 245-281, 1995.

[6] P. Black and D. Wiliam. “Assessment and classroom learning, Assessment in Education”, 5(1), 7-74, 1998.

[7] G. Gibbs and C. Simpson. “Conditions under which assessment supports students’ learning”. Learning and Teaching in Higher Education 1, 3-31, 2004.

[8] D. Nicol. “Principles of good assessment and feedback: Theory and practice”. From the REAP International Online Conference on Assessment Design for Learner Responsibility, 29th-31st May, 2007.

[9] D. Nicol. “Assessment for learner self-regulation: Enhancing achievement in the first year using learning technologies”. Assessment and Evaluation in Higher Education, 34 (3) pps 335 -352, 2009.

[10] I. Stamouli and M. Huggard. “Object Oriented Programming and Program Correctness: The Students’ Perspective”. Icer’06, September 9–10, Canterbury, United Kingdom. ACM, 2006.

[11] M. Taras. “The use of tutor feedback and student self-assessment in summative assessment: towards transparency for students and for tutors. In: Assessment and Evaluation in Higher Education, vol. 26, no. 6, pp. 605-614, 2001.

[12] B. K. Hofer, S. L. Yu and P. R. Pintrich. . “Teaching College Students to Be Self-Regulated Learners”. In: Schunk, D. H. & Zimmerman B. J. (Eds.) Self-regulated learning- from teaching to self-reflective practice (pp. 57–85), New York: Guilford, 1998.

[13] M. Yorke. “Formative assessment in higher education: Moves towards theory and the enhancement of pedagogic practice”, Higher Education, 45(4), 477-501, 2003.

[14] D. Boud. “Sustainable assessment: rethinking assessment for the learning society”, Studies in Continuing Education, 22(2), 151-167, 2000.

[15] D. R. Sadler. “Formative assessment: revisiting the territory”. In: Assessment in Education, 5(1), 77-84, 1998.

[16] R. Ivanic, R. Clark and R. Rimmershaw. “What am I supposed to make of this? The messages conveyed to students by tutors’ written comments”, in: M.R. Lea and B. Stierer, (Eds) Student Writing in Higher Education: New Contexts. Buckingham, SHRE/Open University Press, 2000.

[17] C. Dweck. “Self-theories: Their Role in Motivation, Personality and Development”. Philadelphia: Psychology Press, 1999.

[18] P. R. Pintrich and D. H. Schunk. “Motivation in education: Theory, research and applications”. Englewood Cliffs, NJ: Prentice Hall Merrillm, 1996.

[19] B. J. Zimmerman and D. H. Schunk. “Self-regulating intellectual processes and outcomes: a social cognitive perspective”, in D. Y. Dai & R. J. Sternberg (Eds) Motivation, emotion and cognition. Lawrence Erlbaum Associates, New Jersey, 2004.

[20] Y. Shang, H. Shi and S. Chen. “An intelligent distributed environment for active learning”. In: ACM Journal of Educational Resources in Computing (JERIC). Vol. 1, No. 2, Summer 2001, Article 4, 17 pages. ISSN:1531-4278. New York: ACM Press, 2001.

[21] G. Lazakidou and S. Retalis. “Using computer supported collaborative learning strategies for helping students acquire selfregulated problem solving skills in Mathematics”.Computers & Education, 54(1), 3-13, 2010.

[22] G. Polya. “How to solve it”. Princeton NJ: Princeton University Press, 1973.

[23] A. Bandura.“Perceived Self-efficacy in Cognitive Development and Functioning”. Educational Psychologist, v.28, n. 2, p. 117-48, 1993.

[24] F. Pajares. “Self-Efficacy Beliefs in Academic Settings”. Review of Educational Research, v.66, n. 4, p. 543-78, 1996.

[25] B. J. Zimmerman. “Self-efficacy: an essential motive to learn”. Contemporary Educational Psychology, v. 25, n. 1, p. 82-91, 2000.

[26] D. J. Stipek. Motivation to Learn: from theory to practice. 3.ed. Englewood Cliffs, NJ: Prentice Hall, 1998.

[27] M. P. Mota, L. W. K. Pereira and E. L. Favero. “Javatool: Uma Ferramenta Para Ensino de Programação”. In: Workshop Sobre Educação em Computação, Belém-Pará. Procedings of XVIII Congresso da Sociedade Brasileira de Computação. Porto Alegre-RS: Sociedade Brasileira de Computação, 2008.

[28] http://moodle.org

[29] M. P. Mota, S. R. Brito, S. R, M. P. Moreira, E. L. Favero. “Ambiente Integrado à Plataforma Moodle para Apoio ao Desenvolvimento das Habilidades Iniciais de Programação”. In: XX Simpósio Brasileiro de Informática na Educação (SBIE 2009). Florianópolis, 2009.

[30] M. P. Moreira and E. L. Favero. “Um Ambiente Para Ensino de Programação com Feedback Automático de Exercícios”. In: Workshop Sobre Educação em Computação - Anais do XVIII Congresso da Sociedade Brasileira de Computação. Belém-Pará: SBC, 2008.

[31] S. P. Springer and G. Deutsch. “Left Brain, Right Brain”. W. H. Freeman And Company, New York, 1985.

[32] J. Stasko, A. Badre and C. Lewis. “Do Algorithm Animations Assist Learning?” An Empirical Study and Analysis. In Proceedings of The Sigchi Conference on Human Factors In Computing Systems (Chi’93). 61–66, 1993.

[33] C. D. Hundhausen, S. A. Douglas and J. T. Stasko. “A Meta-Study of Algorithm Visualization Effectiveness”. Journal of Visual Languages & Computing, 2002.

[34] J. Urquiza-Fuentes and J. Velázquez-Iturbide. “A Survey of Successful Evaluations of Program Visualization and Algorithm Animation Systems”. Trans. Comput. Educ. 9, 2 (Jun. 2009), 1-21, 2009.

[35] B. Price, R. Baecker and I. Small.“An Introduction To Software Visualization”. In Software Visualization, J. Stasko, J. Domingue, M. Brown, And B. Price Eds. Mit Press, Cambridge, Ma, 3–27, 1998.

[36] O. Kulyk, R. Kosara, J. Urquiza-Fuentes and I. Wassnick. “Human-centered visualization environments”. Lecture Notes in Computer Science, vol. 4417, Chapter Human-Centered Aspects. Springer-Verlag, 13–75, 2007.

[37] T. L. Naps, G. Rößling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger and J. A. Velázquez-Iturbide. “Exploring the role of visualization and engagement in computer science education”. SIGCSE Bull. 35, 2, 131-152, 2003.

Redesigning Core Programming Courses Through A Direct Instruction Approach

Lethia Jackson, Velma Latson and Monika Gross
Department of Computer Science
Bowie State University, 14000 Jericho Park Road, Bowie, MD 20715, USA

Abstract - *Learning from examples and repetition has been used in K-12 instructional strategies to teach complex skills as mathematics and skills used for the various science disciplines. The use of examples in teaching creates the kind of learning experiences that leads to the successful transfer of what is learned in one distinct problem to a different problem. Student success in the gatekeeper courses in the Computer Science Department is heavily dependent on the students' ability to transfer knowledge from one course to the sequent course. A Direct Instruction (DI) approach to teaching is used to address cognitive learning and information transfer in the computer science gatekeeper courses.*

Keywords: programming, online, tutoring, direct instruction, learning environment

1. Introduction

The Computer Science faculty at Bowie State University has continually reviewed the course content delivered in two gatekeeper core programming courses: Computer Science I (COSC 112) and Computer Science II (COSC 113). Students taking these courses continue to struggle with the cognitive domain of learning as well as transfer. The key characteristics of cognitive learning and transfer [1] include three elements:

I. Initial learning is necessary for transfer;
II. Knowledge that is overly contextualized can reduce transfer; abstract representation of knowledge can help promote transfer; and
III. All new learning involves transfer based on previous learning.

COSC 112 is the first programming course in the course sequence, and students may or may not be familiar with the process of computational thinking [8] [7]. The introduction to the material may cause students to experience cognitive overload. Cognitive overload can have a negative impact on the students' ability to transfer knowledge. Therefore, cognitive load theory [3] assesses the memory load required to perform a task for a learner by examining the memory limits of verbal and nonverbal memory workspace. Three assumptions stand in cognitive load theory [3]:

1. Temporary memory used for reasoning comprehension and learning has limited capacity;
2. Relatively permanent memory is unlimited in capacity; and
3. The mechanics of cognitive processes decreases the temporary memory load.

Consequently, if temporary memory is limited and overloaded, knowledge, comprehension, an approach to an outcome, retention and transfer of learning will have negative results. Cognitive load theory poses three independent sources of cognitive load [3]:

1. Temporary memory is required to complete a task;
2. Irrelevant information that may or may not contribute to learning how to complete a task are presented to the learner; and
3. Useful information is used to improve learning on how to complete a task.

To address the struggles of the students, the faculty modified the present learning environment for both programming courses (COSC 112 and COSC 113) to a Direct Instruction [6],[9] teaching approach which incorporates the following aspects: a) mandatory tutoring b) repetition and c) selected programming concepts based on "key characteristics of learning and transfer" [1].

Direct Instruction, as defined by [6], is "an instructional model that focuses on the interaction between teachers and students that is comprised of modeling, reinforcement and successive approximation." The following includes the components of DI [6]:

- Lesson is delivered in small parts.
- Student outcomes and objectives must be stated clearly.

- Students have opportunities to apply their new knowledge of concepts and terms with previous knowledge.
- Students practice concepts as introduced.
- Students continue to practice concepts through group work or independently that promote transfer of a distinct problem to a different problem.
- Instructors provide feedback to students at every practice opportunity.

2. Project Description

The Computer Science Department redesigned both COSC 112 and COSC 113 to a direct instruction-learning environment in an effort to increase students' cognitive domain of learning as well as transfer. Two sections for each course participated in the course redesign. The course redesign included three objectives:

I. Prevent cognitive overload.
II. Give content questions repeatedly to meet all levels of thinking complexity (knowledge, comprehension, analysis, application, and synthesis).
III. Increase transfer of information from one distinct problem to a different problem.

In objectives I and II, students taking COSC 112 and COSC 113 in spring 2010 were given a pre-test, three tests, weekly quizzes, and a final exam. Each programming course had designated topics that had to be comprehended before taking the next subsequent course in the sequence. The pre-test and the final exam were the same. The pre-test was administered on the first day of class. The purpose of the pre-test was to assess whether students had any prior knowledge to the course content. The answers on the pre-test were measured against the answers on the final exam to assess information learned during the semester.

The final exam was comprehensive. Each question on the final exam addressed all of the designated topics and content discussed in the course. Three questions from the final exam were disbursed among the test and quizzes. The three questions were labeled as structured question 1, structured question 2, and structured question 3. Faculty teaching COSC 112 and COSC 113 met to discuss which three questions would be selected as the structured questions. Each of the three structured questions met the following criteria:

- Content of the question is necessary to prepare students to comprehend material covered in the next course sequence.
- Content of the question is known as difficult for the student to grasp.

Each of the three tests covered the content taught within the period prior to the test. Each test also contained one of the three structured questions. For example, test 1 contained several questions that pertained to the first four weeks of material taught during the course. One of the questions on test 1 was structured question 1. Test 2 contained several questions on material taught in weeks 5 through 8. One of the questions on test 2 was structured question 2. This process continued for Test 3. Each structured question was assessed to measure whether students had improved their knowledge of the content after reviewing the question a number of times.

To assess the lesson segments, weekly quizzes were given. The quizzes were announced and covered the material taught prior to administering the quiz. Quizzes were not given during the week of the test or final exam. The faculty teaching the core programming courses agreed to nine quizzes for the semester. The quizzes contained no more than two questions. A quiz that contained a structured question was considered to be a structured quiz. The purpose of the structured quiz was to assess that particular question. The results of the quiz revealed whether students needed more instruction for that particular topic. There were four weeks of instruction before each test. The structured quiz was given anytime within that four-week period. The structured quiz was not necessarily given the week prior to the test.

Table 1: Course Schedule

Week	Tutoring	Assessment	Structured Question
1		Pre-Test	
2	Standard	Quiz 1	
3	Structured	Structured Quiz 2	#1
4	Standard	Quiz 3	
5		Test 1	#1
6	Standard	Quiz 4	
7	Standard	Quiz 5	
8	Structured	Structured Quiz 6	#2
9		Test 2	#2
10	Structured	Quiz 7	#3
11	Standard	Quiz 8	
12	Standard	Quiz 9	
13		Test 3	#3
14	Standard		
15		Final Exam	#'s 1, 2, and 3

Table 1 gives an example of the course schedule during the semester.

In objective III, weekly mandatory tutoring worth at least 10% of the students' grade was given.
Tutoring was delivered either face-to-face or online. Students received online tutoring via software available through Angel called Elluminate. The online tutoring session had to be delivered and verified by the instructor of the course. Elluminate is Web 2.0 application conferencing software used to promote virtual collaboration and e-learning environments. Learning is enhanced when communication tools are used to initiate and sustain exchanges among participants [5], [4]. Some structured tutoring occurred using Elluminate to demonstrate the skills necessary to solve the problems. Students were able to attend each virtual session and each session was recorded. Students were able to access the recorded session whenever they needed to review the material if they were not able to attend the virtual session. Technology included interactive technology, which allowed students to become actively involved in the learning process [2]. This gave the students flexibility since tutoring was mandatory.

Tutoring was categorized as either standard tutoring or structured tutoring. Standard tutoring allowed the student to lead the tutoring session with questions concerning any topic of their concern. Structured tutoring allowed the instructor to augment standard tutoring by focusing on the topic of the structured question. Before a structured quiz, students received structured tutoring.

3. Pre –Test Results

The assessment of the pre-test revealed that none of the students in COSC 112 had prior knowledge to the content of the course. The pre-test assessment for COSC 113 showed that a few students had prior knowledge. The assessment of the pre-test answers as compared to the final exam answers revealed that students did comprehend most of the content delivered in both courses.

4. Results of the Structured Questions

A structured question earned a grade of 100 if the question was completely correct. Structured questions that received partial credit were not counted. If students received a 100 on a structured question, it indicated that the students comprehended the content of the question. Based on the structured quiz assessments, the results revealed that structured tutoring regardless of delivery did assist the students in comprehending the content material. The test results showed that a repeat of instruction after the quiz improved test results as compared to quiz results of that same structured question. The results of the final exam fluctuated compared to the test and quiz results for any particular structured question. The assumption is the overwhelming aspect of a final exam.

Table 2: Student outcomes for Quiz/Test/Final

Tutoring	Course Section	Quiz/ Test/ Final	SQ #1	SQ #2	SQ #3
Online	COSC 112 - Eve				
		Quiz	0	5%	0
		Test	63%	33%	5%
		Final	80%	37%	5%
Online	COSC 112 - Day				
		Quiz	33%	52%	7%
		Test	66%	88%	71%
		Final	50%	93%	42%
F2F	COSC 113- Eve				
		Quiz	25%	62%	0
		Test	31%	36%	31%
		Final	38%	50%	43%
Online	COSC 113- Day				
		Quiz	5%	5%	0
		Test	24%	33%	5%
		Final	32%	37%	5%

Table 2 gives the percentage of students that received a grade of 100 on the structured question per quiz/test/final.

5. Summary

A Direct Instruction teaching approach proved to be beneficial in addressing cognitive learning and information transfer in COSC 112 and COSC 113. Students that consistently participated in the tutoring performed better than those that did not. All weekly quizzes were announced and given after the students had received tutoring. Students that consistently came to class and took the quiz also performed better in the class than those that did not. The weekly quizzes helped the students to comprehend information in chunks and to focus on that particular content. Administering weekly quizzes prompted the students to study throughout the week. The quiz scores show that some students did comprehend the material after a tutoring session. After the quiz, the content material was discussed and reviewed as part of the class lecture.
At the time of the test, the students had already seen the structured question a number of times via tutoring, quiz, and additional class lecture on that particular content. The

test was given after three quizzes where one quiz was considered a structured quiz. The test scores confirm an improvement in comprehension of the content as compared to the quiz scores.

Students had been tested on the structured question a number of times before the final exam. Based on the results of the final exam, the percentage of students passing the structured question is variable. All of the final exams scores improved over the quiz scores. For the most part, most of the final exam scores demonstrated improvement or remained the same as compared to the test scores.

6. Conclusion

The results of the Direct Instruction approach to teaching indicate that the students retained information and performed well in the course. According to PeopleSoft data, the number of sections offered fall 2010 for COSC 113 increased by one. In addition, the number of sections offered for the next subsequent programming course after COSC 113, COSC 214, increased by one section. An increase in the number of sections offered indicates retention of the students majoring in computer science or computer technology.

7. References

[1] Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000).How children learn. *How people learn expanded edition: brain, mind, experience, and school*. (pp. 79-113).Washington DC: National Academy Press.

[2] Brown, J. S. (2000). Growing up digital: how the web changes work, education and the ways people learn. [Electronic version] *Change* Mar/Apr.

[3] Doolittle, P.E., McNeill, A. L., Terry, K. P., & Scheer, S. B. (2005). Multimedia, cognitive load and pedagogy. In S.Mishra & R.C. Sharma (Eds.), *Interactive multimedia in education and training.* (pp.184-212). Hershey: Idea Group Publishing.

[4] Hannafin, M., Land, S. & Oliver, K. (1999). Open learning environments: foundations, methods, and models. In C. M. Reigeluth (Ed.). [Electronic version] *Instructional-design theories and models (volumes II)* (pp 115-140). Mahwah, NJ: Erlbaum.

[5] Jonassen, D. (1999). Designing constructivist learning environments. In C. M. Reigeluth (Ed.). [Electronic version] *Instructional-design theories and models (volumes II)* (pp 115-140). Mahwah, NJ: Erlbaum.

[6] Magliaro, S., Lockee, B., & Burton, J. (2005). Direct instruction revisited: a key model for instructional technology. *Educational technology research and development, 53* (4), 41-56.

[7] National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking.* Washington DC.

[8] Papert, S. (1993). Mindstorms: children, computers and powerful ideas. New York: Basic Books.

[9] Wilson, B.G. & Myers, K.M. (2000). Situated cognition in theoretical and practical content. In D.H. Jonassen & S.M. Land (Eds.). *Theoretical foundations of learning environments.* (pp 57-88). Mahwah, NJ: Erlbaum.

INDEMAJ: A research-based evaluative report on the 1st pilot project on using technology to support at risk learners in a government school in an Emirati rural area

Dr. Eman Gaad[1]

ABSTRACT

The purpose: The purpose of this paper is to report and summarise the research-based evaluation of the 1st pilot project undertaken by Indemaj, a non-profit organisation that establishes Recourse and Development Centres (RDCs) in government schools in rural areas of the UAE. The DRC supports learners who are at risk of dropping out of the education system through the use of technology.

Design and methodology approach: Research methodology was mostly of a qualitative nature. These consisted of data collected from interviews, observations supported by video recording for evidences, and questionnaires distribution and collections for analysis. For the purpose of validity extensive data triangulation (Robson 2002:174, 5; Rudestam and Newton 2001:100) was used during the data collection and analysis phases of this research where varied sources were used in collecting of data related to this RDC research.

Research limitations and implications: As this the first project that is used as a pilot the researchers were well aware of a series of limitations including bias, the lack of historical record of previous similar projects in the country, and the complexity of the roles of the researchers. However, the use of triangulation and checking allowed for raising the standards of validity. The implications were overwhelming as the research showed the RDC not only achieved what it was intended for but explored many issues to be followed on for future projects. The values which the RDC in this rural area brought were beyond educational and far from curriculum based. The RDC proved to have an impact on the whole school especially on students with special educational needs. It created an original unique social and educational experience that made the school like no other.

Originality/value: This paper will be of interest to those involved in education of children with special educational needs and in particular, those involved in the role of NGOs in developing education in rural areas. It will be certainly of interest to those interested in observing the changes in the education system within the Arabian Gulf area especially in the UAE.

Keywords Special Educational Needs, technology in the classroom, Educational development, Rural areas, Innovations, United Arab Emirates

Paper type General review/research report

INDEMAJ: A research-based evaluative report on the 1st pilot project on using technology to support at risk learners in a government school in an Emirati rural area

Introduction:

The United Arab Emirates (UAE) is located at the southern tip of the Arabian Gulf. The neighbouring countries are Saudi Arabia, Qatar and the Sultanate of Oman. The total area of the UAE is 83 600 km. It has a tropical desert climate with very little or no rainfall. The UAE is governed by a federal system founded on 2 December 1971. The union is formed of seven emirates: Abu Dhabi, Dubai, Sharjah, Ajman, Umm al-Quwain, Ras al-Khaimah and Fujairah. Abu Dhabi city is the capital of the UAE (Camperpix, 1998, p. 5).

INDEMAJ is a newly established organisation under the Honorary Royal Patronage of HRH Princess Haya Bint Al Hussein, Wife of HH Sheikh Mohammed Bin Rashid Al Maktoum, Vice President and Prime Minister U.A.E. and Ruler of Dubai. INDEMAJ has been set up with the mission to support the UAE Ministry of Education in its mission to ensure the delivery of quality education to each and every child attending UAE Government schools. INDEMAJ aims to achieve this objective through providing Resource and Development Centres, Teachers, Training, Mentoring, Monitoring, and research. All schools will be eligible for projects but there will be special focus on schools in the rural areas of the UAE.

A Memorandum of Understanding has been signed between the UAE Ministry of Education and INDEMAJ in order to create a partnership that will assure sustainability of the objectives of the projects.

[1] Author of correspondence: Dr Eman Gaad eman.gaad@buid.ac.ae
Gaad is the Dean and a Senior Lecturer in Special Education, British University in Dubai.

INDEMAJ has been offered and are presently negotiating partnership with universities for research and teachers training purposes. Funding for the projects is raised through collaboration with corporate organizations and sponsorships.

The 1st project:
For the pilot project, INDEMAJ tied up with Khadeeja Bint Khoailed (KBK)Primary School for Girls in Hatta. Hatta is a town in the Hatta mountains in the north of the emirate of Dubai. It is about 115 km east of Dubai City. Because of its altitude, Hatta has a milder climate than the city of Dubai, making Hatta a popular vacation place for Dubai and other UAE residents as well as tourists seeking a quiet getaway time. The task was to set up a Resource and Development Centre (RDC) in a specially allocated room within the school premises, and did joint planning with the management for the training and mentoring. A typical RDC includes varies resources to enhance learning in different ways that appreciates multiple intelligences, different learning styles, and interests. Most of the resources also can be used for different educational, social, and behavioural multiple levels purposes. A smart electronic board and a software screen reader package (known as Ibsar) are included alongside many games, books, materials, and tools suitable for cycle 1 primary level.

KBK School for Girls was established in 1990s but refurbished under the care of HRH Princess Haya Bint Al Hussein in 2006. It was set to provide education for children from the neighboring area age 3 -11 (KG to fifth level – 1st cycle). The Kindergarten department caters for boys and girls while the primary section is for girls only. The school has 375 enrolled pupils with 7 students in the special classroom.

The school is reasonably resourced and seems very organized and students have a good national discipline record. It is designed in a modern way where the management has visual access to almost all classrooms in the one level school. It is the one out of three schools under the Dubai education zone that has a special classroom, and a KG department. They also have a gifted club that provides very innovative materials including CD library with all lessons on PP presentation format.

In order to support social inclusion for children with special needs a link will be established with a nearby special needs centre where children from the centre will be invited to attend the RDC and learn along with the children in the mainstream school once a week. It is also being considered to use the RDC for adult literacy classes where the technology of the Smart Board and the specialized Ibsar software will help to enhance learning.

With each project established in a government school is such rural area is to offer a training package for teachers at the selected school. Such package is set to equip users of the RDC to make maximum use of it and research-based evaluation package for mentoring purposes to follow INDEMAJ's progress and the effectiveness of its projects. A typical training package would include an induction session for staff and a series of break out professional development sessions.
The three hour induction session includes introductory session about the role of the RDC in the school, and the community as well as its use for different purposes, learners, and to serve different teachers and subjects. How a whole school approach is needed to make maximum benefits of the RDC is also touched upon.

The research questions:
This report will be answering the two main primary questions in order to report the overall use of the RDC:
Q1: What is the quality of learning that the RDC offers to its users?
Q2: To what extent is the RDC achieving its intended objectives in the school?
<u>Other secondary questions include:</u>
Q1: How are the resources in the RDC used?
Q2: How is the technology in the RDC used?
Q3: What is the effect of the RDC on students' social values?
Q4: What is the effect of the RDC on the status of the RDC Facilitator and the relationship with the teachers at the school?
Q5: What is the effect of the RDC on parental perception of the school?
Q6: What is the link between the RDC and the school's values, policy, mission, and vision?

Rational and significance of the evaluation:
During the past few years, educators have experienced a steady flow of change in the composition of their classrooms and in the responsibilities required to meet the needs of their students (Khan 2005). Special educational needs are extremely common in the UAE and yet there have been relatively few population based studies in childhood in the UAE and most of these are not focused on such children in the mainstream setting. INDEMAJ is an organization that seeks to help all individuals to reach their full potentials. INDEMAJ pays great attention to research-based studies as stated in its mandate. Researching the pilot project in particular was a must for this new organization that is formed from a board of trustees from various backgrounds but united in the cause and in the interest of developing education in rural areas in the UAE. Researching how the teachers are using the RDC to deliver what it was intended and established for was also essential for INDEMAJ's future practice. INDEMAJ is seeking through this research to understand to what extent teachers as users of the RDC are helping all learners to reach their full potentials. According to Moffett (2000), teachers need to be sensitive to the educational needs of students with SEN, and utilize strategies such students need to learn, if they are to be provided with the most appropriate educational services.

In line with Petty (2004) who argues that "Each learner is unique and has individual needs. If the needs of our learners are discovered and met, the chances of success (referring to educating) are greatly increased" (2004:496). Cooper extends this thought when he comments that a current view expressed by "psychologists and educationists" is that "effective teaching always involves creating circumstances that cater for the characteristics of individual children" (Cooper 1999:25). However when research previously, Alghazo and Gaad (2004) study on the Emarati (nationals) mainstream teachers revealed disturbing findings that teachers in UAE government schools felt students with special needs lack skills needed to master the mainstream classroom course content. Therefore it is essential for such a budding and caring organisation like INDEMAJ to examine to what extent it affect those who are targeted and intended to be helped while educated in the mainstream school to narrow and bridge the gap in essential skills needed to be able to achieve in the government mainstream school.

Methodology and data collection:
INDEMAJ's research methodology was mostly of a qualitative nature. Mertens (1998) defines the purpose of qualitative methods in research as providing "an in-depth description of a specific program, practice or setting" (1998:159). INDEMAJ used non-participant observations to observe closely how the RDC is used in action. Observation was used as in the process of education and educating, diverse and complex activities and/or interactions occur within a classroom resulting in a lot of observable activity at any given moment (Wragg 1999:2, 5) which becomes a source of considerable collectable data (Cohen et al. (2000:147). Mertens supports this sentiment stating "that a considerable amount of work occurs during the data collection and analysis phases of a qualitative study" (1998:348, 9). Additional to this there are the complexities and intricacies of individual children to be considered. Therefore the choice of methodology for INDEMAJ's research was based on qualitative methods which are by nature more descriptive of finer details in the reality of everyday occurrences than Quantitative data would be (Robson 2002:455). Quantitative data however were only used when appropriate to analyse questionnaires responses related to the effectiveness of the training delivered by INDEMAJ to support the use of the RDC. Furthermore for the purpose of validity extensive data triangulation (Robson 2002:174, 5; Rudestam and Newton 2001:100) was used during the data collection and analysis phases of this research where varied sources were used in collecting of data related to this RDC research. These consisted of data collected from interviews observations supported by video recording for evidences and questionnaires distribution and collections for analysis.

Observations were on a non-participant basis in agreement with Silverman (2003:233-248) who argues in favour of being non responsive in the classroom and thereby avoiding influencing as far as possible of the observation through direct interaction, while still acquiring descriptive data of actual occurrences and behaviour (Robson 2002:310- 312).
In an effort to control reliability of data collected, data was corroborated via cross referencing where possible (Robson 2002:101) with information collected from other sources. Further more interviews conducted individually with the special education class teacher, the RDC facilitator, the school principal

and in the case of children interviews were in the form of focus group interviews with children as prime users of the RDC.After each period of observation and interview a clear written account of the data observed was compared to the audio recorded file for clarity and accuracy of content before the latter was labelled and digitally stored2.

Limitations and challenges:
There were many explored issues, limitations and/or barriers related to possible bias, ethics and ethical considerations relevant to this research. Prior to data collection permission was sought from the school in Hatta and obtained from the participants. While protection of personal privacy would be respected and strived for, assurances given that anonymity was far from easy. This is due to the sole role of facilitator or the principal for example it was impossible to insure anonymous account. In order to respect participants rights and maintain ethical standards every effort was made to adhere to all other ethical considerations (Mertens and McLaughlin 2004:151). The findings were presented in a detailed report which this document represent a summarised version of it. There was optional access for comment made available to all participants.

INDEMAJ's researcher faced several ethical issues within this research. First and foremost there is the issue of potential bias and conflicting roles on one of the authors part as a researcher, teacher/trainer, INDEMAJ's board member, mentor to the facilitator and observer. This is related to accuracy and objectivity and raises the question, in what part will the author will be reflecting as the researcher due to the numerous possible roles she found herself in? (Walford 2001:79). However, on a positive note the researcher was able to utilize her 'insider' view of being a board member and a person who was very involved in the design, and the establishment of the RDC. This happens when she used to reflect on and evaluate the literature and current RDC with a high level of motivation and understanding of the subject which could only come from having lived in the whole situation as a person who is 'too involved'.

Another point of concern was related to the consistency of the observed data collected where access was the limiting factor as it is not possible to predict in advance how many lessons or even full days the researcher will be able to be present at school based on timetable, willingness of teachers, teachers who use and who do not use the RDC and other internal school factors. This is because up to the time of writing this report the agreed timetable was not fully implemented due to factor that INDEMAJ has no control of. Therefore and despite the prior planning over the phone with the facilitator the researcher was obliged to schedule observations over a short interval of time whereas data collected over an extended period in a possible new future or follow up study may prove more representatives of the effectives of the RDC.

Research findings:

Findings from interviews:

What is the quality of learning that the RDC offers to its users? This was the main purpose in mind when interviewing with various users of the RDC. Participants confirmed that the RDC provides opportunity for enhanced learning.
Comments like:

> *'The teacher will be no longer transferring the knowledge in the RDC'*

And ….

> *'The RDC is helping the teacher to be flexible as the resources are varied to meet the needs of all users*'

Learners are quoted as finding learning in the RDC as a whole new learning opportunity. It opens the doors for self learning which transfers the concept of the child as a recipient of knowledge to a seeker of knowledge that are used in practical situations. The facilitator puts it beautifully:

> '*self learning. The RDC is encouraging research it is like a whole new world of learning that is new, exciting and certainly more effective'. 'What strikes me is the girl herself,… now she can use the varied material for self learning."*

The special education classroom eco this by stating:

[2] A summarized account of each interview was produced and was gathered in Appendices but due to the word limit was not included. Copies of the full version of the report that included extensive detailed data evidences could be obtained on request from Indemaj by request.

"This differs from what I used to do before. I used to dig out pictures from the 3computer and illuminate it, now the girl get the picture from the smart board in seconds and she pulls, twist, enlarge and minimise ..this is a whole different experience".

A science teacher who took her 5th grade class into the centre for a science lessons states on the issue of quality of learning:

"I love it and the children love it here. Although it is new and require a lot of training we find the learning is simply booming here. Children relate to science ..my subject .. better and I sense the results."

Interviews helped to answer the second research question:

To what extent the RDC is achieving its intended aims in this school?

While the RDC was praised for its flexibility and appreciation of the diverse needs in diverse subjects, certain subjects were stated to benefit more than others:

"Arabic language will have limited benefit unless the teacher uses the pictures in the smart board …Science .. Maths are certainly the best and the English too".

Regarding achieving the intended aims staff commented on how the RDC is *'more'* helpful to children with varies special needs and learning disabilities.

Findings from observations:

This observation took place in the classroom during the mid morning of a school day (full video transcript of observational account supported with pictures is available on request from the authors). The observation showed clearly that the teacher is aware of the various areas of the RDC, and how to use them for teaching and revision. The teacher seems to be very confident in the use of different shapes for example on the smart board, and the girls seems confident to use the board, and were not scared to try new options on the board while the teacher was encouraging them and do and 'undo' if they made a mistake. This increased and focused participating from the girls, and the spirit was very high in the RDC. By and large the girls seem to be enjoying themselves in the RDC while learning and revising important topics, and learning new skills. From the researcher's own account of the observation the following extract shows how successful the RDC is in meeting its aims and objectives. Self esteem was increased, taking turns, collaborative work, interest in technology, mastering ICT skills, and above all learning according to one's needs and potentials were among the values the RDC brought to the school:

"The teacher wrote some text on the board, and the girls came up one by one to the board whenever they could suggest some verbs. It was clear that the girls know how to use the board."

From the following extract it is also clear that girls with special educational needs in this school were overcoming many challenges that they face in the traditional classroom:

"The girl made a mistake and the teacher told her to undo, and the girl knew what to do. She then re-wrote the verb on the board. The teacher asked the girl to change the colour to distinguish her work from other text on the board which she did."

Girls took turns to come out to the board and write verbs which they had suggested. The teacher was in full control of the classroom. It was also clear that the teacher is in full command of the smart board knowing how to use the board to achieve the revision aims. She used the board to de-differentiate verbs. The Ibsar software was well used and certainly attracted the attention of students for long period of times. Values like collaborative work, peer tutoring, team management, assigning tasks, and above all the positive enjoyment of learning were clearly visible.

Analysis of findings from training related questionnaires

Two kinds of forms were distributed to a total of 24 participants during one of the training sessions. There was a 100% return rate as we asked them to complete the forms straight after the session. One was on the attitudes towards the RDC and how it will help teachers to meet the needs of all learners and the second was on the quality and the content of the training sessions. Participants had to comment on a five point scale (options varied from strongly agree to strongly disagree). Participants also had an open ended question to add any comments they wished to.

Questions on the first one included:

Attitudes towards the RDC and how it will help teachers to meet the needs of learners in their classrooms (please refere to App 1:
2nd set of evaluation forms included questions related to the quality of the training sessions:
The feedback was positive in general and comments from the open-ended question included:
"I am more aware of varies types of special needs in my classroom than before."
"I knew that different kinds of learners learn differently, but I think the RDC will help everyone."
"I am excited about the smart board and am looking forward to learn and incorporate technology in my teaching."
"The curriculum may not allow time to make the most of the RDC."

INDEMAJ and Inclusion:
Internationally there has been an increased awareness of children's rights, including the right of access to education (Daniels 2000:69,71; Engelbrecht et al.1999:47) with increased focus on the rights and needs being met of children with learning difficulties and/or disabilities(Thomas and Loxley 2001 cited in Brusling and Pepin 2003: 197). In regard to the latter the concept of inclusion arose. Inclusion means different terms to different people, however could be broadly defined as:

> 'A broad concept dealing with educational access, support for learning' and 'equal opportunities for all pupils, whatever their age, gender, ethnicity, attainment and background'. "
>
> (OFSTED 2000 cited in Jones and Smith 2004:115)

Currently a new Federal Law for special needs was published (Federal Law No.29 for the year 2006) for special needs rights. Based on it educating special needs students is the responsibility of the MOE. Article no.12 stated that " the Country should grantee an equal chance in educating students with special needs within all educational organizations ………, in regular classes or in special classes if it is needed" (Federal Law No.29/2006, Article No.12). That obligates all schools to open their doors for every student with any special needs. At present the Federal Law according to observer is 'on hold' and everyone is awaiting implementation/adoption, or reinforcement. According to sources at the Ministry of Education, the Ministry now is working towards translating the Federal Law to a set of written rules where students with special needs will be accepted in any school without conditions. So the Law currently is in a state of 'hold' until decision makers work out exactly what will be done with children with special needs.

While INDEMAJ's RDC does not fully represent the above concept of full inclusion in UAE schools in rural areas it can be seen as a stepping stone once the government of the UAE decides what inclusion means to them, and how it will adopt a full inclusive policy.

The RDC is designed to allow all students to reach their full potentials. Various educational games resources, materials and assistive technologies are used to allow for vary learning styles to achieve this aim (students reaching their full potentials). Petty however, states that although a student may have a preferred style of learning, this should not be used exclusively as "multiple representations of the concept" being taught would "allow them to experience it in different ways" which are all add to the overall understanding (2004:140, 1).This statement is based on his argument that "it used to be thought by some that learning styles were equivalent and interchangeable routes to the same destination and that students should use the road that suits them." Whereas "the emerging view is that each style offers a different perspective on the learning and that they are complementary and mutually supportive rather than interchangeable" (2004:151). He continues raising the possibility that this approach enables a student to work on areas of weakness while benefiting from stronger areas. So a mixed approach between what the teacher wants to achieve (the lesson/curriculum goal) and what the child wants to learn and how s/he wants to learn it is currently adopted and should be always in mind when the RDC is used.

Discussion and Recommendations
A recommendation is defined by Hornby as "the act of telling somebody that something is good or useful, the best thing to do" (2000:977). In this regard information from all the involved participants in the RDC is taken into account when considering and stating possible recommendations.

It is worth mentioning that based on the following recommendations, and with a revised strategic plan, Indemaj developed substantially and went on to cover 10 schools nationwide in UAE rural areas. The 1st

report showed that all interviewed participants (teachers, management staff, and children) are in agreement with that the cheerful environment in the RDC has an impact on the behaviour and learning of users. Interviews considered that children's environment as an intrinsic part of them and the RDC had an effect on them.

Due to all of these views it would be recommended and even necessitated that children's environment receives considerable more attention in order to fully understand its significance in their learning and ensure that its negative impact is minimized while its positive attributes are known and raised to facilitate their learning experience. Ware defines a "responsive environment" as one in "which people get responses to their actions, get the opportunity to give responses to the actions of others, and have the opportunity to take the lead in interaction" (2003:1). He argues that this type of environment benefits and even enables in some cases "social, communicative and intellectual development" (2003:1, 4). Therefore the RDC environment should be accessible to ALL children in the school and the timetable that allows all children in the school to use the RDC should be adopted as soon as possible.

Social interaction for the main users (children from the resource room and special education classroom) is limited to occasional mixing in the corridors or hall/playground during school assembly and break type gatherings. The RDC was the subject of such interaction. Therefore in order to encourage development of social skills across age and abilities it is recommended that a communal recess be implemented as soon as possible and a mix group of children attend the RDC. Again the school need to work on a timetable that allows mix abilities children to take lesson inside the RDC. Originally the suggested timetable from INDEMAJ would allow all children to use the RDC as part of their regular routine as well as the social inclusion of the children from the next door centre for special needs to visit and use the RDC. The research shows more emphasis on mixing children from various abilities should give and the RDC is the perfect place for such opportunity. Teachers now get full training on hand on skills inside the RDC.

Wragg (1999:96) states: "In any curriculum which is negotiated rather than imposed, both teacher and pupil may need to act more flexibly and responsibly to each other." Flexibility is required in helping deliver diversity in lessons differentiated while striving to maintain a broad exposure to the curriculum for users in order to keep their interest and prevent boredom setting in especially as the RDC becomes a 'normal' everyday practice for those in the special classroom and the recourse room. This requires frequent training by the facilitator and continuous support from experiences teachers (those who have been there and done that as already users of the RDC) and those who are just about to start the training to become users. I suggest the immediate start of a training schedule to maintain a balance and equal opportunity for all teachers to become RCD users so we don't wait to train while the children are actually in the RDC. Observation[4] showed that confident teachers have an effect on the enthusiasm and self esteem of their students. I suggest the school take advantage of the post exam and pre summer holiday period to run a schedule to train all teachers to be users of the RDC. I also suggest an interest club or group meeting once a while to update all staff with new skills they learnt and applied with their children in the RDC to insure transferring experiences and to keep the wheel of innovation going on in this dynamic school.

As for the students who are enrolled in the special classroom and those who attend the resource room I strongly recommend that the RDC and its wide variety or resources to be incorporated in their IEP (Individualised Educational Plans). The RDC should be seen as everyday practice not a luxury 'lemon green' room that children go as a 'prize or reward' when they are good and deprived from when they are bad[5]. I finally suggest adding to the induction something about experiences from the pilot project, what worked, what didn't and what could be learnt from the pilot but keeping in mind that each school is a unique dynamic institution with its own identity and variables.

Conclusion

This report attempted to answer the two main primary questions to report on the overall use of the RDC:
Q1: What is the quality of learning that the RDC offers to its users?

---

[4] Transcripts of observations are available on request from the author of correspondence due to word limit guidelines.

[5] Transcripts of interview with the special classroom teacher are available on request from the author of correspondence due to word limit guidelines.

Q2: To what extent the RDC is achieving its intended aims in this school?
Regarding the first question research showed that the quality of learning that the RDC offers to its users is outstanding, and enriching on multiple levels. As for the second question it was also clear that the RDC is achieving its intended aims in this school to a very good extent. Moreover other desirable values and educational ethics were achieved through the RDC.

In addition, the research also was set out to answer some secondary questions such as:
On *how the resources in the RDC are used*, the research proved that the resources in the RDC are being well used, but the whole school approach needs to be adopted better to insure ALL children have access to the RDC in next academic year. On *how the technology in the RDC is used*, the technology is the most used resource in the RDC. The effect of the RDC on students' social values was positive and encouraging. It was certainly bridging the gap between those who attend the special classroom, the resource room, and the non-disabled children in the school. Self esteem, enthusiasm, familiarity with technology, and innovative and creative thinking was also noted and increase via the use of RDC in the school. On *what is the effect of the RDC on status of RDC Facilitator and relationship with teachers at the school*, the RDC had a positive effect on the status of RDC Facilitator and the relationship with the teachers at the school. She is now seen as a mentor and as a gateway to this 'new and innovative' initiative at the school. Regarding the effect of the RDC on parental perception of the school, It certainly had a positive effect on parents and they now perceive the school as a modern school and a home for the new technology. Finally there are enough evidences that the RDC is in line with school values, policy, mission, and vision.

REFERENCES

Alghazo, E. and Gaad, E. (2004). General Education Teachers in the United Arab Emirates and Their Acceptance of the Inclusion of Students with Disabilities. British Journal of Special Education, 31 (2), 94-99.

Camperpix (1998)

Spectrum Guide to United Arab Emirates .Nairobi, Kenya: Camerapix Publishing International.

Cohen, L., Manion, L. and Morrison, K. (2000) Research methods In Education. (5th Ed). London: RoutledgeFalmer

Cooper, P. (Ed) (1999) Understanding and Supporting Children with Emotional and Behavioural Difficulties. London and Philadelphia: Jessica Kingsley Publishers.

Gaad, E. (2001). Educating Children with Down Syndrome in the United Arab Emirates. British Journal of Special Education, 28, (4), 195 – 203.

Gaad, E. (2004). Cross-cultural perspectives on the effect of cultural attitudes towards inclusion for children with intellectual disabilities. International Journal of Inclusive Education. 8 (3), 311-328.

Hornby, A.S. (2000) Oxford Advanced Learner's Dictionary.6th Ed. United Kingdom, Oxford: Oxford University Press.

Khan Lavina L. (2005) A STUDY OF PRIMARY MAINSTREAM TEACHERS' ATTITUDES TOWARDS INCLUSION OF STUDENTS WITH SPECIAL EDUCATIONAL NEEDS: A PERSPECTIVE FROM DUBAI, unpublished MEd dissertation, British University Dubai.

Mertens, D. (1998) Research Methods in Education and Psychology. California : Sage Publications.

Mertens, D. and McLaughlin, J. (2004) Research and Evaluation Methods in Special Education. United States of America: Corwin Press, Inc.

Moffett, C. (2000). Sustaining change: the answers are blowing in the wind. Educational Leadership, 57 (7), 35-38.

Petty, G. (2004) Teaching Today. (3rd Ed). United Kingdom : Nelson Thornes Ltd.

Robson, C. (2002) Real World Research. (2nd Ed).Oxford : Blackwell Publishing.

Scott Velda, V (2006) Peters' day: The Education Experience of a child with Lennox Gastaut Syndrome, unpublished Med dissertation, British University, Dubai.

Silverman, D. (2003) Investigating Qualitative Data: Methods for Analysing Talk, Text and Interaction. 2nd Ed. London : Sage

Walford, G. (2001) Doing Qualitative Educational Research. London: Continuum.

Ware, J. (2003) Creating a responsive Environment for people with profound and multiple learning difficulties. Great Britain: David Fulton Publishers Ltd

Wragg, E.C. (1999) An Introduction to Classroom Observation. 2nd Ed. London and New York : Routledge Falmer.

Collaboration using Social Networks for Team Projects

Kendall E. Nygard, Lisa Bender, Gursimran Walia, Jun Kong, Kanwal Gagneja
Department of Computer Science
North Dakota State University
Fargo, USA
{kendall.nygard, lisa.bender, gursimran.walia, jun.kong, kanwalinder.gagneja}@ndsu.edu

Marvin LeNoue
American English Institute
University of Oregon
Eugene, USA
mnlenoue@uoregon.edu

***Abstract*—A controlled study was conducted to evaluate the potential benefits of a social networking tool for collaboration within teams while carrying out a major course project. The study involved thirty-one teams that carried out term projects in a Social Implications of Computers course. The work of each team required social and ethical analyses that pertain to a computing or internet related topic. Some of the teams utilized the Ning social networking tool as the primary communication and collaboration vehicle and others used more conventional means, such as face-to-face meetings. The degree of diversity among team members was measured using survey methods and controlled in forming the teams to establish a quasi-experimental design. Statistical analyses support the hypothesis that the Ning system was helpful in elevating the performance of teams with high diversity.**

Keywords-social networking; team projects; diversity

I. INTRODUCTION

Team projects are highly valued components of many courses within most university programs. It is well understood that team projects help individuals to develop soft skills that will be critical in the workplace, such as respectfully and professionally listening, contributing, writing, presenting, brainstorming, discussing, and assisting others. In addition, learning important concepts and perspectives in a topic area new to them is facilitated through working in teams. At North Dakota State University (NDSU), we have introduced team projects into a course entitled Social Implications of Computers. Unlike team projects in a capstone or other computer science course, these projects involve deep exploration of social and ethical issues in a chosen topic area as well as detailed analyses of the ramifications of alternative action choices. Working on team projects of this type is a new experience for most of the students and the maturity of their thinking skills is advanced in important ways.

Social networking systems (SNS), such as Facebook, are wildly popular in today's world. This popularity and the availability of many interaction features in SNS suggest that they could offer new and powerful possibilities for technology-enhanced education. Our study specifically examines the potential for a social media platform called Ning to provide helpful mechanisms for student teams to employ for communicating, cooperating, and collaborating while carrying out a major course project

The members of project teams can vary widely in terms of dimensions such as age, gender, cultural background, intelligence level, motivation, and experience in the working world. Highly diverse teams face special problems in conducting their projects. More specifically, heterogeneity heightens the potential for teams to face difficulties in establishing a leadership structure, mutual respect, open discussion, motivation, and generally getting best efforts from all of the participants. Our study compared the effects of heterogeneous teams with social networking support against the homogeneous teams without social networking support.

The remainder of this paper is organized as follows: Section II provides a general background of social networks in education and discusses problems faced by team members in their class projects. Section III details the design of the experiments that were conducted to investigate the usefulness of social networking systems as a collaboration method for performing the project activities. Section IV presents an analysis of the results from the experiment. Section V gives a summary and the conclusions.

II. BACKGROUND

In the Computer Science and Management Information System programs at NDSU, the course Social Implications of Computers is required for senior level students. A course of this type is often taught within programs that are nationally accredited by the Accreditation Board for Engineering and Technology (ABET). A major team project that develops and examines multiple viewpoints of an ethical issue in computing or the internet is required in the Social Implications course. The project has four deliverables. Because such projects have a long history of being required in the course, guidelines for teams to follow and the grading rubrics for each deliverable are detailed and well-developed. There are typically three people on a team. In many cases, the students on a team do not know each other personally prior to the course, yet they are expected to join forces and work effectively as a team on the course project.

A. Course Projects

Many degree programs in computing education require a substantial development project in which the students work in teams and are fully involved in the analysis, design, and development of a software product. Typically called capstone projects, there is often sponsorship by a company that provides in-kind mentoring, commitments and support. Capstone projects tend to focus on following the formalisms of project management that have evolved over time. These formalisms include such things as guidelines for developing a charter, building a schedule, managing change, following standards, and producing various technical documents that pertain to requirements, design, testing, etc. Social factors of importance in working within teams, such as negotiating, discussing, and communicating are also important in the management of capstone projects. However, Capstone projects are quite different from the ethics projects that we require in the Social Implications course. In the ethics projects, student teams are charged with identifying and analyzing ethical questions, evaluating their importance, researching and understanding the viewpoints of multiple stakeholders, and carrying out analyses that consider the ramifications of possible courses of action. The types of ethical dimensions that are examined as the pertain to computing systems include: a) Quality of life, b) Use of power and control, c) Risks and Responsibility, d) Property Rights, e) Privacy, f) Equity and Access, and g) Honesty and Deception.

We submit that team functioning has strong analogies with the ethical topics being carried out in projects. This is fundamentally why we require that students adopt and represent multiple viewpoints in their project work. More specifically, the social norms that develop for working in teams are similar to the resolving of ethical dilemmas in the topic of choice. Thus, team projects on ethical topics have the double benefits of students learning to work well in teams, reinforced by deep understanding of ethical issues.

B. Social Networks in Education

Social media is a term that applies to networked tools, services, and applications that support webs of interactions among people and with data as well as providing for the creation of online content [1,2]. Examples of social media systems include Wikipedia, MySpace, Facebook, YouTube, Second Life, Flickr, Twitter, and Ning. Social media applications give Internet users the capability to create, contribute, communicate, and collaborate in the online environment without need for specialized programming knowledge. In the spirit of Marshall McLuhan [3], it is understood that social media create communities of people who follow new types of interaction patterns and for whom the pace of their communication lives is altered. Educators are now challenged to create and sustain learning opportunities that leverage these new capabilities.

Ning is a free-form, online platform that allows people to create and host their own social networks. With Ning, users can customize their profile pages, share documents, RSS feeds, comments, photos and videos, chat, post events and post status updates on Twitter and Facebook. Within social media, we find social network systems (SNS) such as Ning, to be particularly useful in creating an open-ended learning environment that provides multiple possibilities for activities and provides students with tools and resources that support problem-solving [4,5].

The use of a SNS a as serious mechanism of education delivery is still in the early stages of development, yet there is evidence of potential in pedagogy [6,7]. Many campuses have committed to Learning Management Systems (LMS), such as Blackboard. However, a LMS tends to focus on content handling and email and is fundamentally teacher/institution centric. When compared with a LMS, social software offers greater interactivity and a distributed web of communication paths. The success of social software is based upon human needs for social interaction and a sense of community [4,8].

Our study is motivated by team problems that we have observed in the past that we hypothesize could be addressed with social networking systems. These include difficulties in students finding times in their schedules to meet regularly in person, situations in which one or more team members emerge as dominant personalities and examples of students who disempowered or disrespected on their team. Diversity on teams amplifies some of these problems.

C. Diversity

Diversity within teams can have effects on how teams conduct their work and set and meet their goals [9]. The primary differences among members of our teams are age, gender, ethnicity, and working experience. There is a tendency for people in teams to categorize one another, essentially distinguishing in-group members (those who are similar) from out-group members (those that are dissimilar) [10]. Work by Brewer [11] and van Knippenberg and Schippers [12] reports on investigations of in-group and out-group interactions. Results indicate that in-group members tend to be favored and trusted over out-group members and cooperation among in-group members tends to be better than among people working in widely diverse groups. Results also indicate that as groups become more diverse, there tend to be more misunderstandings due to the lack of a common frame of reference. Thus, in-group (homogeneous) teams tend to function more smoothly than out-group (heterogeneous) teams. Social networking tools act as an interaction buffer which allow users to continue to interact, but in a less direct way. Users can take the time to thoughtfully post content and take more time assessing and reacting to information that is placed in the SNS. In this way, SNS such as Ning can help to neutralize the potentially negative effects that heterogeneous teams can experience due to the lack of a common frame of reference.

Our study is empirical work focused on examining the use of the Ning social networking system on the performance of heterogeneous teams in completing a semester-long project on a topic with significant social/ethical issues. Our results indicate that using the Ning tool for managing the class projects was beneficial.

III. Study Design

The study is designed to answer the question of whether the use of the Ning social networking system is helpful in addressing problems faced by heterogeneous teams in conducting projects. The study basically compares the performance of student teams that used Ning with the student teams that did not use Ning.

The experiment utilizes a Non Equivalent Pre-test Post-test Control group quasi-experimental design consisting of a control group (consisting of homogeneous student teams that could meet face-to-face and use traditional communication methods such as emails and attachments) and an experiment group (consisting of heterogeneous student teams that were asked to use the Ning tool) to help coordinate and manage their group projects and to facilitate communication among team members. The details of the study are given below.

A. Hypotheses

The following two hypotheses were posed in the study

Hypothesis 1: The *heterogeneous* student teams using the Ning tool produced project quality at least as high as the *homogeneous* student teams that used traditional project coordination methods such as face-to-face meetings, emails, and posted attachments.
Hypothesis 2: Subjects favored the use of the Ning tool for managing the class projects.

B. Independent and Dependent Variables

The experiment manipulated the following independent variables:

- ***The project management technique.*** Subjects either used Ning or used traditional communication and coordination methods for managing their group project activities.
- ***The pre-test.*** Measures were collected from each subject regarding their age, gender, nationality, background knowledge and skills, personality type, work culture, and their preferences for alternative project coordination methods. The results from the pre-test helped form *homogeneous* teams and *heterogeneous* teams.
- **The *training procedures*.** Measurements of the perceived usefulness of training procedures for each student were gathered.

The following dependent variables were measured:

- ***Team performance.*** These measures include the team grades on the final project report and the final project in-class presentation.

C. Participating Subjects

Ninety-two graduate and undergraduate students enrolled in the Social Implications of Computing course at North Dakota State University participated in this study. The ninety-two subjects were divided into thirty-one teams (thirty teams had three students each and one team had four students). The course required that the students work collaboratively on a major project involving a multi-vocal response to an ethical issue related to the world of computer technologies.

D. Artifacts

Each of the thirty-one teams produced a project proposal document for a computer-related topic with ethical issues as shown in Table I. The key concept was that each individual in the group was required to adopt a distinct viewpoint that corresponded to a stakeholder. Rather than writing as a single voice trying to answer the ethical question posed in the proposal, the group members represented diverse views on their topic. The consequences of alternative actions were traced through and evaluated with regard to their impact on other stakeholders.One goal of having the students working together as a team was to help them to *understand* the perspectives other than their own, to respect those perspectives, and enrich and mature their own viewpoints through appreciation of others. The students met throughout the semester on their approved topics and subsequently produced an interim status report, final oral presentation and final report.

TABLE I. TEAMS AND TOPIC TITLES

Team Number	Homogeneous Teams	Heterogeneous Teams
1	Children on the Internet	Use of Computers and Web by Restaurants
2	Computers in Professional Sports	Deep Packet Inspection
3	Global Economy	Devices to Assist People with Disabilities
4	Copyrights in Music and Movies	Identity Theft
5	Technological Responses to Terrorism	Alternate Realities / Virtual Worlds
6	Web in Schools	Computing and Network Access in Kenya
7	Ethical Issues in Airline Security	World After 20 Years
8	Technology in Sports	Computers in Law Enforcement
9	Spam	Free Software and Open Source Software
10	Deep Packet Inspection	Spam
11	Privacy on the Web	Information Welfare
12	Addiction to Video Games	Unmanned Air Vehicles
13	Digital Distraction	Violence in Video Games
14	Facebook – Pros and Cons	Privacy for Organizations and Businesses
15	Music and Video Copyright	Distance Learning
16	X	Copyright Ethics in the Age of Technology

E. Experimental Procedure

To evaluate the hypotheses posed in Section III.A, the study was contained a control group and an experiment group. Figure 1 provides an overview of the procedure followed and Table II shows the timeline.

The details of the experiment steps are given below.

***1) Step1(Pre-Test)**:* A fifteen item pre-study questionnaire administered at the beginning of the semester collected

TABLE II. PROJECT DELIVERABLES AND TIMELINE

Project Deliverable	Project Proposal	Interim Report	Final Report	Project Presentation
Timeline (2010)	March 10	April 9	May 7	April 28 – May 7

information regarding each subject's age, gender, nationality, background knowledge and skills, personality type, work culture, and their preference ofproject coordination methods. The result from the pre-test was used to develop homogeneous teams and heterogeneous teams in Step 2.

***2) Step 2 (Forming Homogeneous and Heterogeneous Teams)**:* The process used for forming student teams is as follows:

a) Assigning point values: Each item in the questionnaire was assigned point values for each of the responses associated with the question, assigning each item approximately equal weight towards determination of a total score for each student.

b) Ranking of Scores: The total scores for all students were ranked and sorted in descending order.

c) Categorizing Student Teams: Two categories of student teams were created: *homogeneous* teams (with very similar characteristics) and *heterogeneous* teams (with diverse characteristics). Since the ideal team size is three, the student responses were divided into tertiles (i.e., groups of high, middle, and lower scores). Students were ranked from highest to lowest within each tertile. For example, in a class of ninety students, the top thirty scores represent the upper tertile, scores thirty-one to sixty the middle tertile, and scores sixty-one to ninety the lower tertile. Teams were assigned by matching ranks across the tertiles. The thirty-one teams were further divided into sixteen heterogeneous teams and fifteen homogeneous teams. The *heterogeneous* teams were generated by randomly selecting subjects from each tertile to form sixteen groups. The remaining students were sorted by standardized score and formed into *homogeneous teams* of students by gathering groups of three as they occurred on the list. Students who are adjacent on the list are of high similarity as scored by our diversity measure.

d) Assigning treatments: The *heterogeneous* teams were asked to use the Ning private social networking system (Ning) to help coordinate and manage their project and to facilitate communication among team members, whereas the *homogeneous* teams were asked to not use Ning, but rather use

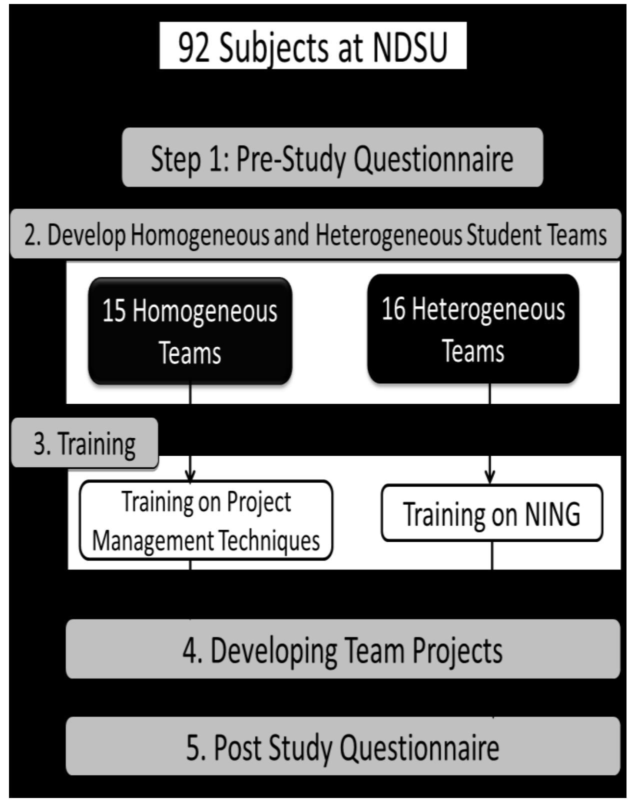


Figure 1. Experiment Design

the more conventional project coordination methods, such as email and face-to-face meetings.

***3) Training**:* Basic training was provided to each group. The *heterogeneous* teams were given hands-on training on how to use Ning to assist them in successfully setting up a Ning site and to utilize the site resources. The *homogeneous* teams were briefed by the instructor on well-established principles for successfully managing team projects. Each training session lasted for one lecture period.

***4) Developing Team Projects**:* The students worked in teams on a semester long project that culminated in a final report and oral presentation from each student team. Most groups chose a topic from a provided list of candidate ideas. Then the teams performed the necessary research to write a project proposal. The research typically included google searches, reading of periodicals and books, personal interviews with subject matter experts, and surveys. Half way through the semester, each team submitted an interim progress report. Near the semester end each team gavean oral presentation on their project and submitteda final written report. Grades were collected after each deliverable (shown in Table II).

***5) Post-Study Questionnaire**:* A ten question post-study questionnaire was administered to the students at the end of the semester. The post-study questionnaire collected feedback from the student teams regarding their percieved effectiveness of their collabaoration methods, communication among team members, and quality of team work in encouraging creativity and innovation.

IV. Analysis and Results

Statistical analyses were carried out on the data collected during the study. The two hypotheses described in Section III.A were tested to determine the usefulness of the Ning tool for collaborating on group projects and student attitudes toward the tool. An alpha value of 0.05 was selected for judging the significance of the results.

A. Hypothesis 1: Comparison of Homogeneous and Heterogeneous Group Performance on the Quality of Projects

This section provides analysis of the effect that Ning had on the quality of projects developed by the teams. The marks scored by the sixteen heterogeneous teams that used Ning and the marks scored by the fifteen homogeneous teams that did not use Ning were compared. Scores for each of the four project deliverables shown in Table II and the overall scores were tested using an independent samples t-test.

Figure 2 shows comparisons of the average scores of heterogeneous teams (HET_{Ning}) and homogeneous teams (HOM_{NONing}) on the *Project Proposal, Interim Report*, *Final Report*, and *Project Presentation*. On the *Proposal* deliverable the HET_{Ning} performed better than the HOM_{NONing}, while the

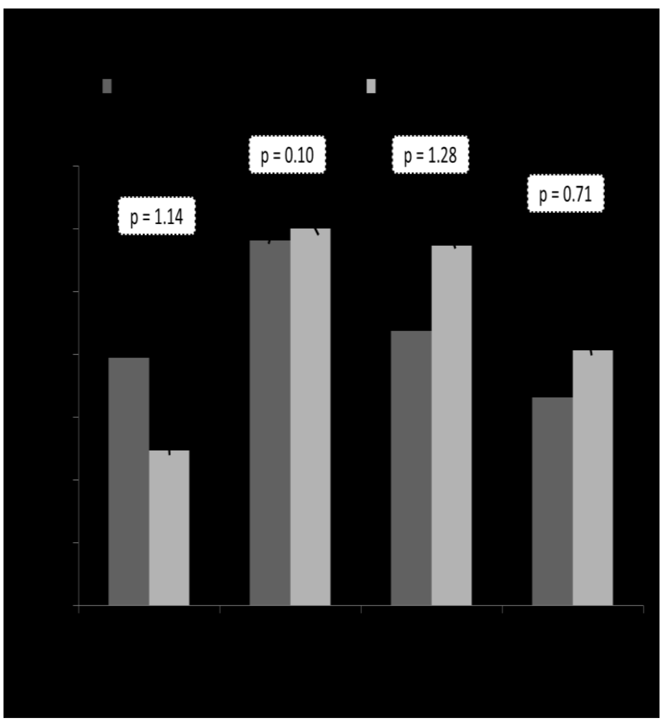


Figure 2. Comparing Performances of Heterogeneous Teams (HET_{Ning}) and Homogeneous Teams (HOM_{NONing}) on Project Deliverables

HOM_{NONing} performed better than the HOM_{NONing} on the others. The p value from the t-tests are illustrated in Figure 2 and do not show a significant difference in the average scores of homogeneous and heterogeneous teams for any of the four deliverables.

In comparing the comparing the total score over all four deliverables, the HOM_{NONing} average score of 181.43 was slightly better than the HET_{Ning} average score of 182.36.

Figure 3 shows a comparison of the total scores of the heterogeneous and homogeneous teams. The results are ordered by increasing number of marks scored by the teams in each group.

An independent samples t-test reveals that the *heterogeneous* student teams performed aswell as the *homogeneous* student teams on their group project (p=0.024). These results indicate that using the Ning tool is beneficial for

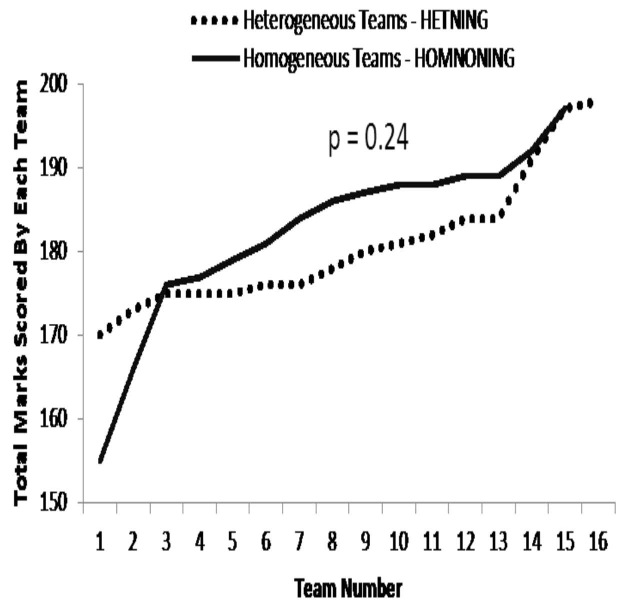


Figure 3. Comparing Total Project Scores of Sixteen Heterogeneous Teams (HET_{Ning}) and Fifteen Homogeneous Teams (HOM_{NONing})

team members with high diversity.

B. Hypothesis 2: Usefulness of the Ning Tool

The usefulness of the collaboration method used by the homogeneous and heterogeneous teams was evaluated using feedback from the subjects in the experiment group (the members of heterogeneous teams) and the control group (the members of homogeneous teams). Nine essential attributes of a collaboration tool for completing a class project were included: a) *Communication among team members, b) Effectiveness of the Team in Organizing project activities, c) Setting and accomplishing project goals, d) Scheduling and attending Meetings, e) Enthusiasm for using Social Networking / Collaboration tools on a project, f) Encouraging creativity and innovation, g) Respect for everyone's ideas, h) Ability to meet deadlines, and i) Effectiveness of the Collaboration Method for completing the class project*.

Forty-eight subjects from heterogeneous teams that used Ning and forty-six subjects from homogeneous teams that did not use Ning rated each of the above nine attributes on a 5 point likert scale (1-Very High, 2 – High, 3- Medium, 4- Low,

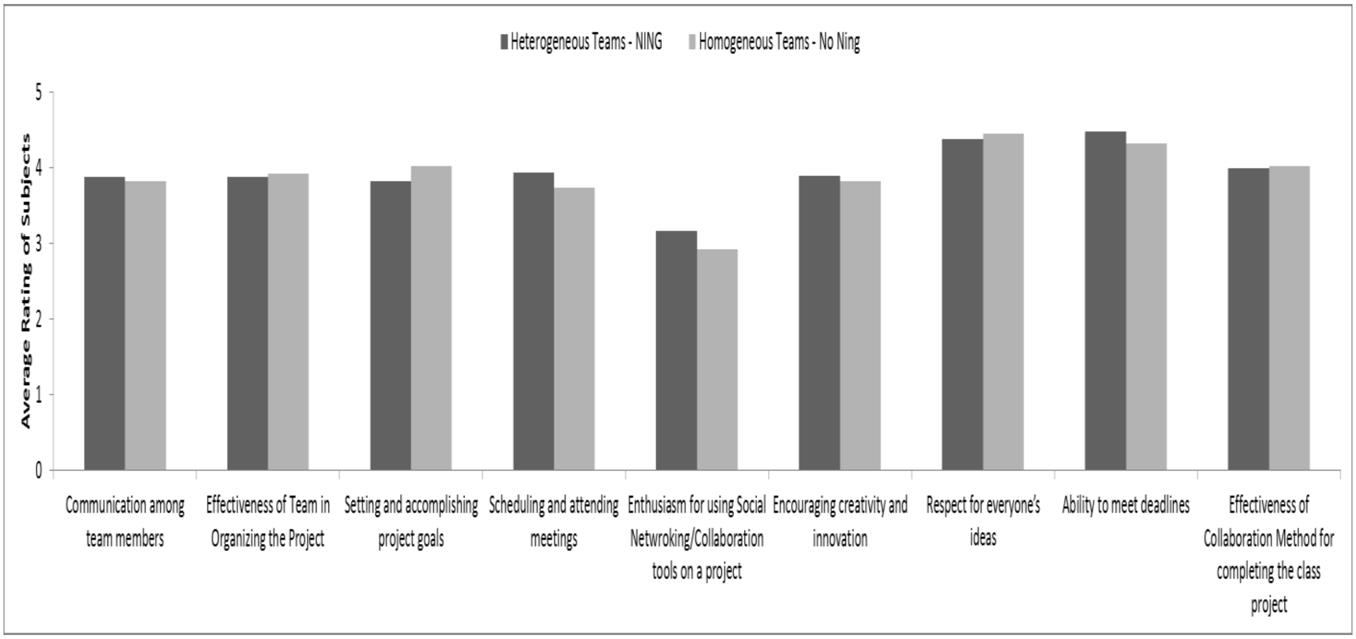


Figure 4. Comparison of Average Ranking of Homogeneous and Heterogeneous Group Members on Nine Attributes

and 5- Very Low). Figure 4 shows comparisons among the average ranking of the experiment and control group subjects for each of the nine attributes. The results show that the homogeneous and heterogeneous teams viewed their collaboration method equally ineffective on all of the nine attributes. So, while the subjects using the Ning tool did not perceive the tool favorably, it was not perceived worse than the traditional communication methods used by subjects in the homogeneous teams.

The results essentially show that neither the Ning tool nor any other collaboration method was viewed favorably for collaborating on the group project. The implications and the discussion of this result are presented in Section V.

V. SUMMARY AND CONCLUSIONS

Thirty-one teams were formed and each team carried out an extensive term project on a computing topic with significant social and ethical issues. Teams with little diversity among their members typically function well on these group projects, while highly diverse teams often have difficulty. Our primary hypothesis is that heterogeneous teams that use the Ning social networking tool as a collaboration platform for their project work can elevate their performance to match that of the homogeneous teams. The experimental study supports this hypothesis with statistical significance. We speculate that the innovative features of Ning that support several types of interaction among people while maintaining them at a distance can be helpful in avoiding breakdowns in collaboration and cooperation that can occur with more conventional meeting methods. Our second hypothesis concerns assertions on multiple attributes for which the students were surveyed. Results indicate that students did not report a particularly favorable attitude toward the use of Ning. We speculate that although Ning was shown to be helpful in stimulating project success and performance, the learning curve for mastering Ning and heavy workload imposed by the project itself dampened the enthusiasm of the students.

ACKNOWLEDGMENT

We thank the students in the Social Implications of Computing course at North Dakota State University for their participation in this experiment. We also thank statistician Curt Doetkott for his assistance with the study.

REFERENCES

[1] D. Boyd, “Taken out of context: American teen sociality in networked publics.” Doctoral dissertation, University of California-Berkley, 2008.

[2] D. M. Boyd and N. B. Ellison, “Social network sites: Definition, history, and scholarship”. *Journal of Computer-Mediated Communication*, vol. 13, Oct, 2007.

[3] M. McLuhan, *Understanding the Media: The Extensions of Man*. New York, NY: McGraw Hill, 1964.

[4] C. Dalsgaard, “Social software: E-learning management systems”. European Journal of Open, Distance and E-Learning. 2006.

[5] S. M. Land and M. J. Hannafin, “A conceptual framework for the development of theories-in-action with open-ended learning environments.” *Educational Technology Research and Development*, vol. 44, pp 37-53, 1996.

[6] C. Dalsgaard. “Social networking sites; Transparency in online education.” Presented at ENUIS 2008 Proceedings, Arhus, Denmark, 2008.

[7] R. Fitzgerald, S. Barrass, J. Campbell, S. Hinton, Y. Ryan, M. Whitelaw, A. Bruns, A. Miles, J. Steele, and N. McGinness, “Digital Learning Communities (DLC): Investigating the application of social software to support networked learning,” http://www.digitalrepository.qut.edu.au/, 2009.

[8] C. Dalsgaard and M. Paulsen, “Transparency in cooperative online education.” *The International Review of Research in Open and Distance Learning*, vol. 10, 2009.

[9] K. Y. Williams and C. A. O'Reilly, "Demography and diversity in organizations: a review of 40 years of research", *Research in Organizational Behavior*, vol. 20, pp. 77-140, 1998.

[10] D. van Knippenberg and M. Schippers, "Work Group Diversity", *Annual Review of Psychology*, vol. 58, pp. 515-541, 2007.

[11] M. B. Brewer, "In-group bias in the minimal intergroup situation: a cognitive-motivational analysis." *Psychology Bullitin*, vol. 86, pp. 307-324, 1979.

[12] D. van Knippenberg, C. K. W. De Dreu, and A. C. Homan, "Work group diversity and group performance: an integrative model and research agenda". Journal of Applied Psychology, vol. 89, pp. 1008-1022, 2004.

Prior Experience as an Indicator of Team Effectiveness in Software Engineering Courses

Karina Assiter
Wentworth Institute of Technology
assiterk@wit.edu

Durga Suresh
Wentworth Institute of Technology
sureshd@wit.edu

Abstract

The purpose of this investigation is three fold: first, to determine how students benefit from learning communities, second, to see if team effectiveness improves with prior team-based experience, and third, to see if there is a correlation between individual student performance and success in a learning community.

The authors intend to study the impact of learning communities on student performance/success by surveying students in multiple sections of a senior year Software Engineering course, where class projects are an integral part of the curriculum, and students are expected to form teams and work together to produce deliverables. These students are in technical majors (Computer Science and Computer Networking) and are proficient in programming languages like C, C++ and Java and usually work on real world projects, such as setting up a website for a small business or creating their own customized project management system. Most of these students enter the workforce right after graduation, and are expected to work in teams, exhibit ethical behavior, have superior communication and leadership skills, and be able to train new hires.

The survey questions focus on how students learn within the scope of their learning community. Summative assessments (course grades are based on individual effort, as well as a students' contribution in the team) are employed to evaluate the student during the course of the team project. Based on the survey results we hope to improve team building exercises and the overall pedagogy of teaching classes that involve teams.

1. Introduction

Collaborative learning "has long been recognized as a effective approach to enhance the learning process" [5]. One of the ways to achieve collaborative learning is by creating learning communities as a part of the engineering curriculum. There are both formal communities created in the form of groups (either by faculty members or by students themselves in order to complete a semester long project) and informal communities, such as study groups and library groups (also called as community- based learning groups) and peer-tutoring sessions. While the method of team formation is important, it is also important to assess the effectiveness of teams once they are formed. The authors of this paper intend to show the effects of the formation of the formal learning communities.

At the authors' institution, team-based collaboration is facilitated by technology, as all incoming freshmen are given a laptop and technology is integrated into the course of study.

When the authors' designed the survey for this research, the motivation was to learn how students work on teams, to see if there is a relationship between working on teams and final grades and to see if prior experience working on teams improves performance (for teams and individuals) in their senior software engineering course (which is also preceded by the students completing two co-ops in industry).

In the next section we discuss work related work to collaborative learning, learning communities and advantages of students working on teams in a Software Engineering (SE) course. Then we discuss our methodology including the presentation of our survey and a description of our cohorts. The results section presents an evaluation of the survey and the outcomes that were discerned; these observations are then summarized in the conclusions section. Finally, we discuss next steps in our section on future work.

2. Prior Art

In this section we discuss the topics that are related to collaborative learning, learning communities and teams in Software Engineering.

2.1 Collaborative Learning

Collaborative learning describes an instructional approach in which students work together in small groups to accomplish a common learning goal. Collaborative learning is based on these principles: (1) Tasks are carefully designed to be suitable for group work. (2) There is positive interdependence. (3) Students are individually accountable for learning and participation. (4) Attention and class time are given to interpersonal/ collaborative skill building. (5) The role of the teacher changes from being "an instructor" to "a guide or a facilitator" [7]. These are important when we consider teams in SE. It is also imperative to understand that even though these principles are not formally specified to the students they are the general principles that we try to follow in order to foster an environment in which students may flourish in a team-based course.

2.2 Learning Communities

Learning communities can be of two types, formal and informal. Learning communities are not a new concept and have been used in various forms to improve student involvement. "Learning communities involve creating a positive learning environment and focusing on interaction between students" [12] and also between students and faculty. Irrespective of how these communities are configured, they have been successful in creating "strategies in helping students adjust to college life and provide support when students face academic hurdles "[9] [10] [11].

2.3 Teams in SE

Software Engineering has always been a hard subject to teach [1]. The importance of incorporating a collaborative learning environment in a project-based course is crucial and is often achieved by the formation of teams. "To prepare students well, they have to learn and gain experience from working in an environment that resembles an industrial situation as closely as possible. We derived the aspects directly from the student's future working environment, i.e., software development in industry." [4]. Team-based projects where students can practice their technical and soft skills are a key feature of many software engineering courses, where the objective is to prepare students for the realities of industrial software development [13].

The growing importance of teamwork in the software engineering (SE) discipline has led to a vast amount of research on SE student team formation [2]. This study is to meant to help us better understand team characteristics and how prior experience in working on a team effects team dynamics, and ultimately a students' final grade in the course.

3. Methodology

Our objective was to evaluate the effect of prior team-based collaboration on student (and team) success in a subsequent course. Thus, we surveyed all current seniors in their Software Engineering sections, where class projects are an integral part of the curriculum and students are expected to form teams and work together to produce deliverables. The survey questions are listed in Table 1.

3.1 The survey

Table 1- Software Engineering Team Survey

Question Type	Ref #	Question
Yes/No	1.	Have you ever worked on a team before?
Integer value	2.	How many times have you worked on a team?
Range of agreement 1=Strongly Agree 5=Strongly Disagree	3	Working on a team (in the past) a was a rewarding experience b. improved my grade c. helped me manage my time better (Time on task!)
	4.	My prior experiences working on teams helped me with a. getting a better grade in the class b. being more effective on this team.
Range of Evaluation 1=It Rocks 5=It Sucks	5.	Team a. Communication b. Work Distribution c. Planning c. Meeting Deliverable Deadlines
Letter Grade	6.	What grade do you expect in the class?

3.2 The Cohort

These students are in engineering majors (Computer Science and Networking) and are proficient in programming languages like C, C++ and Java and usually work on real world projects, such as setting up a website for a small business, making a mobile application, creating a game by integrating open source game engines or creating their own customized project management system. Most of these students enter the workforce right after graduation, and are expected to work in teams, exhibit ethical behavior, have superior communication and leadership skills, and be able to train new hires.

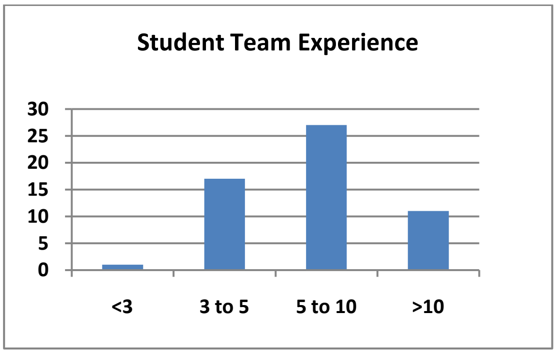
This survey was administered to seniors in the computer science department in four sections. The seniors were from the software design and development class, where teams were comprised to three to four students in self-selected teams.

5. Results

In this section we break our discussion into: prior team experience, evaluation of prior experiences, evaluation of current team and student perceived correlations between prior team experience and current success (both individual and team-based).

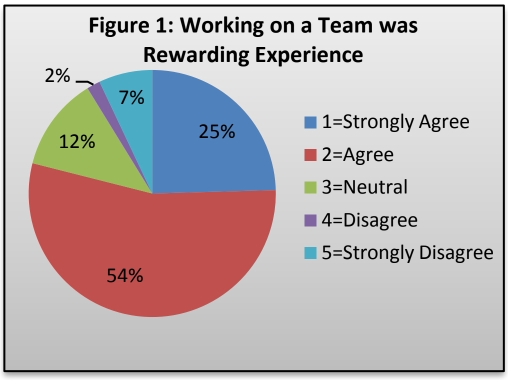
5.1 Prior Team Experience

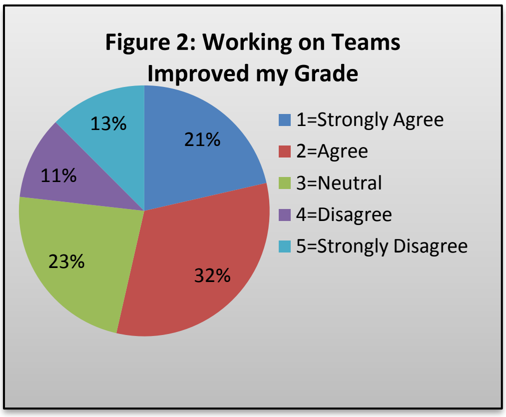
The first result, that is both the least interesting and the most revealing, is that 100% of all respondents had worked on teams in the past. In the following table we then see a breakdown of the actual numeric value that students had entered for the number of times that they had worked on teams:

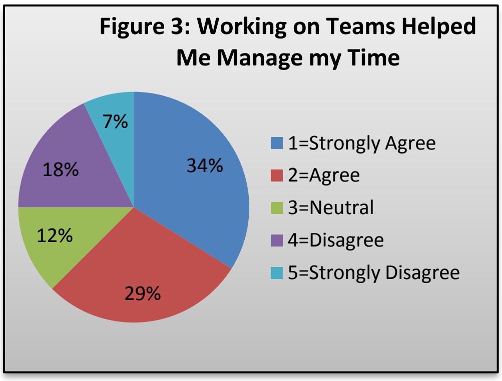


5.2 Evaluation of prior experiences on team(s)

This next set of results provides an valuation of students' prior team experiences. In **Error! Reference source not found.** we see that more than ¾ of all students agree that working in their prior teams was "rewarding". In addition, a little more than half of all respondents (Figure 1) felt that working on prior teams improved their grade. The final figure in this section (Figure 2 and 3 shows that almost ¾ of all students agree (most strongly) that team experience helped them manage their time better.

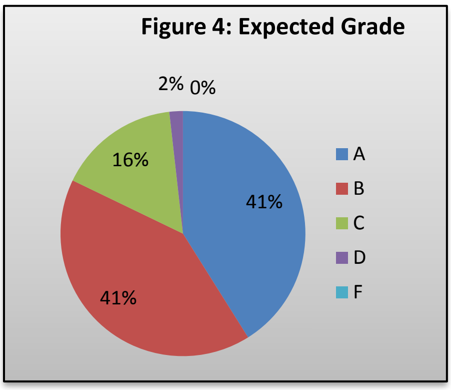






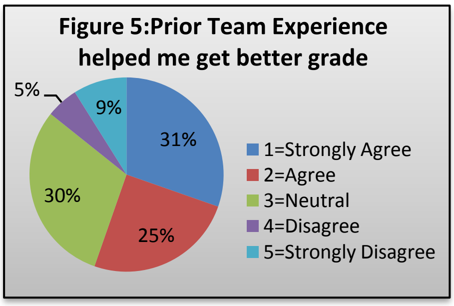
5.3 Evaluation of Current Team

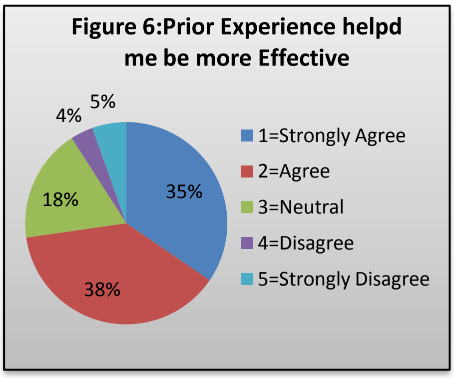
In Figure 5 we see that 82% of all respondents expect at least a B in the course. This result is not surprising, since a majority of the seniors are experienced with their respective professors' team-based assessment rubric's (and, thus, know what to expect).



5.4 Evaluation of Prior team experience on current performance

Though we could look for correlations indirectly by evaluating students' expected performance versus their team experiences, we decided to include questions that would directly correlate prior experience (on teams) with current performance. As we see (Figure 5), more than half of all respondents agreed that prior team experience helped them get a better grade in the current course, and more than ¾ agreed (Figure 6) that it made them more effective on their current team. It is interesting to note that the respondents' effectiveness on a team does not always translate to their expecting a better individual grade.





6. Conclusions

From our study, we found that:

- All students had had prior team experience.
- Students agree that working on teams was a rewarding experience
- 50% of all respondents felt that prior team experience improved their grade.
- 75% of the respondents said that prior team experience helped them manage their time better
- 82% of all respondents expect a B or better for their final grade in the course.

7. Future Work

From this study we have discovered that when students have prior team experience they: manage their time better, get better grades and also feel that working on teams is a rewarding experience. Thus, our future work will investigate further the effectiveness of these prior experiences on current teams. Specifically, in relation to their prior team experience, how has it helped students: manage their time on their current team, master specific software tools that aid in better team communication, and improve their individual and the team's performance *(i.e., grades).* Then, also, what particular aspects of team work do students find "rewarding" and why.

8. References

[1] Mead, N, Carter, D., and Lutz, M. The State of Software Engineering Education and Training. IEEE Software, 14, 6 (Nov/Dec 1997), 22-25.

[2] Lewis, T.L.; Smith, W.J.; , "Building software engineering teams that work: The impact of dominance on group conflict and Performance outcomes," *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual* , vol., no., pp.S3H-1-S3H-6, 22-25 Oct. 2008. doi: 10.1109/FIE.2008.4720498

[3]Devlin, M.; Phillips, C.; , "Assessing competency in undergraduate Software Engineering teams," *Education Engineering (EDUCON), 2010 IEEE* , vol., no., pp.271-278, 14-16 April 2010. doi: 10.1109/EDUCON.2010.5492569

[4] Louwarnoud van der Duim, Jesper Andersson, and Marco Sinnema. 2007. Good Practices for Educational Software Engineering Projects. In *Proceedings of the 29th international conference on Software Engineering* (ICSE '07). IEEE Computer Society, Washington, DC, USA, 698-707.

[5] Nicole Anderson and Chi-Cheng Lin. 2009. Exploring technologies for building collaborative learning communities among diverse student populations. In *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education* (ITiCSE '09). ACM, New York, NY, USA, 243-247.

[6] Antonio P. Volpentesta and Nicola Frega. 2007. Developing a blended-learning community in a university setting. *Int. J. Web Based Communities* 3, 2 (June 2007), 134-150.

[7] Johnson, D. W. & Johnson, R. T. (1999). Structuring academic controversy. In S. Sharan (ed.) Handbook of cooperative learning methods. Westport, CT: Praeger Publishers.

[8] Bailey, R.; Shoffner, M.; Rowan-Kenyon, H.; , "Special session — Integrating learning communities into engineering curricula," *Frontiers in Education Conference (FIE), 2010 IEEE* , vol., no., pp.T4A-1-T4A-2, 27-30 Oct 2010.

[9] Hurd, S. N., & Stein, R. F. Building and sustaining learning communities: The Syracuse University experience. Bolton, MA: Anker Publishing Company, 2004.

[10] Fink, L. D. Creating significant learning experiences: An integrated approach to designing college courses. San Francisco: Jossey-Bass, 2003.

[11] Tinto, V. Leaving college: Rethinking the causes and cures of student attrition (2nd ed.). Chicago: University of Chicago Press, 1993.

[12]Howies, T.; , "Work in Progress - Learning Community and Active Learning Study," *Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual* , vol., no., pp.F4G-1-F4G-2, 10-13 Oct. 2007

[13] Stan Kurkovsky. 2008. Four roles of instructor in software engineering projects. In *Proceedings of the 13th annual conference on Innovation and technology in computer science education*(ITiCSE '08). ACM, New York, NY, USA, 354-354.

SESSION

PROGRAM, CURRICULUM, AND COURSE DEVELOPMENT + ASSESSMENT AND RELATED ISSUES

Chair(s)

TBA

An Industry Focus Group Forum for Engineering Program Assessment

K. Ferens and M. Friesen
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada R3T5V6
ferens@ee.umanitoba.ca, Marcia_Friesen@umanitoba.ca

Abstract - *This article presents a process, results, and experiences of conducting an Industry Focus Group forum for curriculum assessment of the undergraduate electrical and computer engineering program at the University of Manitoba. The forum resulted in the identification of 21 gap subject areas, where a gap subject area was defined as the difference between industry's desired attributes of a new graduate and the actual abilities of a new graduate, at the time of entering the workforce. The identified subject areas were mapped to the Canadian Engineering Accreditation Board's graduate attributes for validation. Proficiency levels were assigned to the CEAB attributes and the identified gap subject areas. Finally, learning outcome statements were formed by assigning Bloom verbs to subject areas based on the selected proficiency levels.*

Keywords: industry focus group, curriculum assessment, CEAB.

1 Introduction

THE Canadian Engineering Accreditation Board (CEAB) has established new (additional) accreditation criteria for undergraduate programs, which, while initially publicized in 2008, will begin to be enforced in 2014 [1]. In contrast to the previous criteria, which specified delivery of minimum hours of instructional content, the addition to the criteria is a specification of 12 learning outcome attributes, which the CEAB feel that new graduates of an engineering program should have acquired from an engineering program.

There are many factors that most likely have motivated CEAB to adopt the outcomes based approach to accreditation. First, as tabulated by [2]: CEAB's graduate attributes reflect the shift made by other accreditation organizations, such as graduate attributes of ABET (formerly, Accreditation Board for Engineering and Technology) in the US in the late 1990s and the associated outcome attributes of the Engineering Criteria 2000 (EC2000); program outcomes of Engineers Australia, specific learning outcomes of the UK Standard for Professional Engineering Competence (UK-SPEC), and topics of the CDIO Syllabus [3].

Second, Canada participates in and has an obligation to meet the requirements of the Washington Accord, which is an agreement between 13 countries to establish substantive equivalence in the accreditation of engineering programs [4]. Accordingly, CEAB's graduate attributes are similarly phrased as the graduate attributes and professional competencies of the Washington Accord [5].

Third, recent studies [5] [6] [7] [8] have identified newly critical characteristics of engineers to meet rapid changes in technology and globalization of markets, such as international cross collaboration of design ability, manufacturing, and the international interweaving supply-chain. The studies stress the importance of educating engineers to prepare them to assimilate the complex technical, social, and ethical questions raised by emerging technologies, and offers recommendations of desired attributes so graduates are better prepared to work in a constantly changing global economy. Accordingly, CEAB's new graduate attributes are phrased in this manner to reflect the changing characteristics of the future engineer: for example, "economic, environmental, cultural and societal considerations" in the design attribute, and the "Impact of engineering on society and the environment" attribute.

Fourth, specification of outcomes (attributes) of new graduates for accreditation purposes is a collection of statements of requirements which a graduate must possess at the time of graduation. Previously, CEAB exclusively used the amount and type of inputs given to students, in the form of instructional course content, which can be seen as design issue, without explicit specification of the requirements. Widely acknowledged, that which is input to a system may not necessarily be exhibited at the output. Accordingly, CEAB has adopted outcome (aka attribute or requirements) based assessment accreditation criteria.

To enforce these new attributes, CEAB's new accreditation criteria require that engineering institutions:

1. Create and demonstrate that they have processes in place to measure quality of their new graduates in the context of the 12 CEAB graduate attributes; and,

2. Create and implement processes to continually develop and improve their programs.

In anticipation of the looming 2014 CEAB new accreditation criteria enforcement date, some wise engineering institutions across Canada already have begun considering how to meet the new requirements. There are two general approaches taken: course level assessment and program level assessment.

Many engineering institutions in Canada and the US have begun with the course level approach, which performs assessment at the course level, and extrapolates the result to achieve a program level assessment [2] [9] [10]. In the course level approach, a subset of courses in a program is selected to take part in the assessment. Instructors of the subset specify outcome statements for their courses. These outcome statements are communicated to students at the beginning and throughout the courses, so that students know what is expected of them. For example, a microprocessing systems course may describe an outcome as “after completing this course, students should have the ability to write an assembly language program to implement a high level language structure, such as for and while-loops.” To implement the outcomes based assessment, instructors develop test questions, assignments, labs, or projects to allow students iterative opportunities to develop and master the knowledge or skill, as well as to measure a student’s ability in the specified outcomes. For a more detailed measurement, a rubric is developed to assess the degree of a student’s performance in the specified outcomes. The outcome statements are mapped to the 12 CEAB attributes. The outcome statements and CEAB attribute mappings are compiled for all courses taking part in the assessment, and the results are presented to the CEAB at the next accreditation visit. The institution argues that course level assessment sufficiently satisfies CEAB’s requirement that students are assessed at the time of graduation, since students will not have forgotten the expected and specified course level outcomes, particularly for a cohesive set of courses that have interwoven prerequisites, and the outcomes acquired in a prerequisite course must be known and applied in successor courses. However, the disadvantage is that students do forget, or at least their retention of the acquired attributes will be lower at the time of graduation, and so the measurements done at the course level will not be indicative to the same degree were the assessments done at the time of graduation.

The program level of assessment is performed after graduation, and is much broader in scope in that it requires an assessment of the program at large, of which the curriculum is one component. The program level assessment is thus more difficult to implement. There are inherent problems with the program level approach to assessment. The future movements of graduates of a program need to be tracked to determine the companies they work for, if any at all. Exacerbating the problem is that some graduates don’t find jobs in the area of their expertise, and some find non-engineering jobs. Furthermore, some graduates take a leave of absence (i.e., go on vacation), and thus their retention of acquired attributes may diminish.

A concomitant challenge in implementing both course- and program-level assessment is the faculty development task, to equip and enable professors to develop credible and justifiable outcome statements for their individual courses, as well to develop assessment tools (assignments, tests, labs, exams) that validly assess the outcomes in integrative ways.

In the program level assessment, stakeholders need to be identified, such as instructors, industry, department heads, deans, students, alumni. Questionnaires or other assessment tools need to be developed for each type of stakeholder, and arrangements need to be made to administer the questionnaires, which would require a great deal of administrative effort. This paper’s contribution is the design and development of a process for implementing an Industry Focus Group (IFG) forum focused on attributes of graduates from the undergraduate electrical and computer engineering program, in order to integrate the input of this stakeholder group into an assessment process. This paper also reports on the results of an IFG meeting which was held on 22 February 2011 in Winnipeg, Canada.

The remaining parts of this paper are organized as follows: Section 2 gives an overview of related work in the area of program level assessment. A detailed description of the design, development, and results of the IFG forum are given in Section 3. Section 4 concludes the paper.

2 Related Work

Conceive, Design, Implement, and Operate (CDIO) [3] recommends and specifies a template industry questionnaire for assessing abilities of new graduates. The questionnaire may be modified to suit a particular engineering discipline.

Cloutier [11] has developed an industry questionnaire, which inquires about the “expected” as well as “observed” levels of ability of a new graduate of an engineering program. He argues that many industry representatives would not take kindly to being subjected to differently phrased and formatted questionnaires from different engineering institutions, each requesting to assess their graduates. Accordingly, he has developed a unified questionnaire, which is “a CDIO-modified scale [3] with formal definitions of levels to ease interpretation and reduce variance; checked it with half a dozen industrial advisors and half a dozen professors, then enlarged the focus-group to 15 industrial supervisors on a student-appraisal-form.”

The difference between this paper and the previously mentioned works is: This paper uses the proficiency levels defined by CDIO (Table 6 of [3]), but, instead this paper reports on a face-to-face meeting with industry representatives, which provided better communication possibilities.

Nair and Patil [12] report on the missing links between engineering graduates and employers’ expectations at Monash University. The paper outlines the findings of a recent survey conducted on the perception of graduate attributes of Monash engineering graduates. A gap analysis was done, based on an importance-satisfaction and 5 point Likert scale.

The ISPE [13] established an industry focus group procedure to identify the needs and expectations of industry regarding skills and subject matter knowledge of recent college graduates seeking jobs in the industry. The process this paper uses is similar to [13], except that the gap subject areas discovered in this paper were mapped to CEAB attributes for validation. Furthermore, this paper differs in that the CDIO proficiency levels were used to determine proficiency levels of the identified subject areas, as well as Bloom verbs were used to create outcome statements.

3 Forum Design, Development, and Results

3.1 Preliminaries

Members of local industries, associated mainly with electrical and computer engineering, such as computer network security, embedded systems, satellite telecommunications, aerospace telecommunications, information communications technology (ICT), utilities, and government were notified by email, requesting their participation in a industry focus group (IFG) for curriculum assessment. The email communicated the purpose of the IFG, which was to get Industry's perspective on the requisite knowledge, skills, and attitudes of a graduate of an undergraduate electrical and computer engineering program, in order to begin a successful industry career. The Industry's perspective will then be used as input to help identify gaps and to modify, shape, and restructure the ECE Curriculum at the University of Manitoba. The forums are being planned biannually to facilitate continual improvement of the program. The letter went on to state that the forum will give participants the opportunity to give back to the University and community and to lend their experience to what local industry values and expects from those who work within it.

Based on the responses received from the invitation, there was a 94% acceptance rate. Most of the rejected reasons were because of time conflicts and other commitments. Based on the accepted responses received from the invitation, a database of contact information, including name, email address, title, position within the company, company name, and educational background, was created.

An appropriate venue was booked, which was located at a central location, not on University campus. The meeting room offered free wireless Internet service, data projector with screen, free parking (with plug in!), water and coffee service, writing pads with pens, and a separate lunch room.

An orientation package was sent to the participants one week prior to the event. The package contained the agenda of the meeting, background description on the new CEAB accreditation criteria, a description of the process to be followed in the meeting, and a timeline for completion.

3.2 Meeting of 22 February 2011

The first meeting of the Industry Forum was held on February 22, 2011. Each participant was given a workbook, which was used throughout the meeting to record responses and data. The session was opened with a thank you to those who had volunteered their time to the effort. The facilitator gave the welcome message and introduced himself. The facilitator then asked each person to introduce themselves, noting their company, professional area of responsibility, and educational credentials.

3.3 Role of Academic Participants

An important message from the facilitator was to differentiate the roles of those attendees from industry (15) and those from academia (3). The meeting's goal was to obtain the perceptions and priorities of industry participants. The primary role of academic participants was to be observers of the process and to clarify any curriculum issues.

3.4 Canadian Professional Organizations

The facilitator gave a brief summary of professional engineering organizations in Canada: Engineers Canada and the role of the Canadian Engineering Accreditation Board (CEAB), highlighting the new accreditation criteria [1].

3.5 Constraints on Learning Outcome Statements

The facilitator explained that the Industry Focus Group's identification of missing or lacking attributes of new grads needed to be encompassed within the 12 CEAB attributes. A learning outcome not classified under one of the 12 CEAB attributes may not be recognized by the CEAB. Learning outcomes not recognized by the CEAB may not count as a measure of accreditation. A program falling short of accreditation measures may be denied accreditation. Graduates of a non-accredited program would not directly be able to apply for licensure to practice engineering, and this may impact local industries ability to hire new grads.

3.6 Seating Arrangements

The participants were seated in a set of four tables, five persons per table. The tables were organized according to discipline and professional level of expertise as much as possible.

3.7 Phase 1: Gap Analysis

Phase 1 of the session allowed participants to identify gaps in knowledge, skills, and attitudes they encounter when hiring new graduates. The entire group took part in this session. The facilitator posed the following trigger question to the industry participants:

"Recall that when new graduates begin to work at your company; orientation, training, or other methods were typically required to establish a knowledge base so that they could begin to function and contribute to the company. For the recent university graduates that you've hired, what were the primary gaps that these new graduates needed to acquire through training, mentoring, or other means?"

The facilitator then led a brainstorming session and recorded all thoughts to a PowerPoint slide Table 1 "Gap Exploration: Ideas." Meanwhile, the facilitator's assistant converted each idea to a subject area on the PowerPoint slides Table 2 "Gap Exploration: Subject Areas."

Table 1. Gap Exploration: Ideas.

#	Missing or Lacking in a New Graduate
1	Database architecture understanding foundations
2	Lack of knowledge of eng standards and how to apply them. do they exist, IEEE
3	Independent critical thinking
4	Do not exhibit a desire to obtain in depth knowledge (behavioural)
5	No concept of project management , change request, testing, verification, requirements development
6	Requirements engineering, verification, validation (systems engineering)
7	Time management
8	Not good enough at written, verbal, presentations communication
9	Lack of understanding practical applications of theory. Not enough lab exposure, practical writing code.
10	Balancing business versus technical needs, spend too much time on technical side and not enough on the business side
11	Cross disciplinary knowledge, like mechanical, electrical, bio, etc.
12	Risk assessment analysis and response, what are the risks of taking a certain approach to a design
13	Structured analysis methods
14	Decision making, using associated tools, under uncertainty, with quantitative outcomes
15	Resume writing interview skills
16	Variability of grad proficiency
17	Why is it important to be registered with APEGM, when to use the stamp
18	Linux kernel drivers, not knowledge on FPGAs (teach the concepts, but not the details)
19	Project management I the control of the process. How to operate under a project management system
20	Understanding the entire business cycle, product requirements, manufacturing, etc
21	Lack of understanding of the real world
22	Societal impact (one course in curriculum)
23	Need more exposure to contract writing as background

Table 2. Gap Exploration: Subject Areas.

#	Subject Area	Votes	Priority
1	Database architectures - fundamentals		
2	Engineering standards (IEEE, ISO, IEC, etc.) – awareness of, application of, limitations of	2	
3	Cognitive skills: critical thinking, self-initiated or independent learning	10	
4	Metacognition (thinking about thinking): assessing one owns learning ("am I done?" "Did I find what I needed?")		
5	Demonstrated desire for in-depth knowledge		
6	Systems engineering	7	
7	Development processes & non-technical aspects (customer requirements, testing, verification, validation, etc.)	6	
8	Project management	2	
9	Communication – written & verbal; email; presentations	6	
10	Application of theory in real world practice; practical, hands-on skills as gained, e.g., at RRC	2	
11	Balancing business needs vs. technical needs; adjusting technical design to other constraints	2	
12	Risk assessment, analysis, response; structured analysis methods	5	
13	Cross-disciplinary understanding: knowing enough about other disciplines as they impact one's work	2	
14	Decision-making tools – quantitative and qualitative	1	
15	Interviewing skills		
16	Variability in graduate knowledge, skills, characteristics: ways to reduce variability?		
17	Professional licensure and regulation		
18	Technical knowledge: representative examples - Software engineering, operating systems, tools that support the technical knowledge, e.g. FPGAs, Linux, kernel drivers,	2	
19	Business functions: design, development, manufacturing, operations, administration, etc.	1	
20	Societal impact; environmental impact		
21	Engineering ethics; engineering law.		

In the first part of the session, each member was given an opportunity to make a statement. Duplicate ideas were appended to the related idea already on the slide. Once the group had run out of ideas, the facilitator reviewed each item and clarifications were made where necessary. The number of gaps was 23, about one per member.

3.8 Alternative Brainstorming Method

The facilitator attempted to extract more ideas from the participants by employing another brainstorming method: each participant was requested to do the following:

1. Think of an answer to the trigger question, no matter how trivial or questionable the answer is!
2. Write the answer down in one or two lines on a sheet of paper, one idea per sheet.
3. Pass the paper to the neighbour on your left to stimulate his or her thoughts.
4. Look at the answers passed from your neighbour to the right and pass these on to your left as well. Use the ideas you have read to stimulate your own ideas.

Moving around each table, ask everyone to read out one of the ideas on the sheets that happen to be in front of them. The ideas were recorded on PowerPoint slides.

3.9 Gap Statement Conversion

After the gap exploration process, the group reviewed the subject areas, which had been compiled by the note-taker. The conversion of gap idea to subject area was done simply by extracting the *nouns* and *noun phrases*. Redundant, vague, or highly general nouns were eliminated. The note-taker passed the subject area PowerPoint slide (Table 2 "Gap Exploration: Subject Areas.") to the facilitator, and this was entered in the facilitator's copy of the presentation. The subject areas were displayed in a table which had three columns: gap subject, number of votes, and priority.

3.10 Subject Area Prioritization

After the group had reviewed the subject area list, each participant was asked to identify the three most important subject areas and write them down in their workbooks. After two minutes, the facilitator counted the number of votes for each subject area by a show of hands. The facilitator entered the number of votes for each subject area and then identified the subject areas which received the top five votes. See PowerPoint slide "Table 1: "Gap Exploration: Subject Areas."

3.11 Lunch and Break (12:00 – 12:45)

There was a 45 minute lunch and break time. Participants were directed to the lunch room, which was located on the main floor. Participants were encouraged to use this informal time to continue to discuss the themes of the morning's meeting.

3.12 Phase 2: CEAB Attributes

The second phase of the meeting asked the participants to map the 21 subject areas to CEAB attributes. Each table was given a copy of the complete list of subject areas, generated during the meeting. The facilitator instructed each group/table to select a note-taker and a spokesperson. A table named "Subject Area to CEAB Attribute Mapping" of the workbooks was used to record the mappings (See Table 4 for a summary). The groups were given 25 minutes to complete their tasks and were told to include a short 10 minute break, if required. During the remaining 5 minutes of this period, the facilitator asked the spokesperson of each table to report their mapping by submitting their workbook pages to the note-taker for transcription.

3.13 Phase 3: Assigning Proficiency Levels

The third phase of the meeting asked the participants to assign required or desired graduate proficiency levels to the 12 CEAB attributes. The facilitator explained that, typically, not all CEAB attributes have the same weight and the same level of importance in their business areas. For example, some companies require their engineers to be more proficient at legal aspects, while others require more proficiency in the design attribute. As well, this phase of the focus group was to highlight that developing mastery in the subject areas is an ongoing professional process that may be initiated during the undergraduate program, but continues during professional practice.

For the mapping, the members were asked to perform the following, individually: to use one of the given proficiency levels (Table 6 of [3]); to assign proficiency level to each of the CEAB attributes; and to enter that proficiency level in their workbooks. After completing the assignment, each member of the table passed their CEAB attribute proficiency assignments to the spokesperson of the table. The spokesperson calculated the average proficiency level for each CEAB attribute, and made this information available to the members of the table. The members then discussed the proficiency levels assignment to come to a relative agreement. Then, the facilitator asked the spokesperson of each table to report the average proficiencies and pass their workbooks to the note-taker for transcription (Table 3).

While the groups/tables were performing the CEAB proficiency level assignments, the note-taker used the mappings reports to create a list of CEAB attributes with each of the subject areas listed beneath each attribute.

3.14 Phase 4: Selecting Bloom Verbs

The final step was to select an appropriate Bloom verb as the desired proficiency level for each of the subject areas. Table 8 of [3] shows the mapping of Bloom verbs to proficiency levels. The Bloom verb selections were collected and summarized in Table 4.

Table 3. CEAB attributes proficiency assignment. Each table (Tn) assigned a proficiency level to the 12 CEAB attributes.

#	CEAB Attribute	Proficiency Level			
		T1	T2	T3	T4
1	A knowledge base for engineering	3	3	3	3
2	Problem analysis	2	4	4	2
3	Investigation	3	4	4	1
4	Design	2	2	2	2
5	Use of engineering tools	3	1	1	3
6	Individual and team work	3	2	2	2
7	Communication skills	4	4	2	3
8	Professionalism	2	3	2	3
9	Impact of engineering on society and the Environment	2	3	2	2
10	Ethics and equity	2	3	2	3
11	Economics and project management	2	2	2	2
12	Life-long learning	3	3	3	3

Table 4. Summary of learning outcomes statements. (Legend: V=Votes, P=Priority, Tn=Table number n)

#	Subject Area	V	P	CEAB Attribute				Bloom Verb			
				T1	T2	T3	T4	Table 1 (T1)	Table 2 (T2)	Table 3 (T3)	Table 4 (T4)
1	Database architectures - fundamentals	0	14	1	1	1	1	Explain	Explain	Explain	Explain
2	Engineering standards (IEEE, ISO, IEC, etc.) – awareness of, application of, limitations of	2	6	1	1	5	1	Interpret	Locate & Classify	Recall	Interpret
3	Cognitive skills: critical thinking, self-initiated or independent learning	10	1	2	2	3	3	Recognize	Practice	Demonstrate	Recall
4	Metacognition (thinking about thinking): assessing one owns learning (“am I done?” “Did I find what I needed?”)	0	14	2	3	3	2	Recognize	Demonstrate	Demonstrate	Recognize
5	Demonstrated desire for in-depth knowledge	0	14	12	12	12	12	Identify	Demonstrate	Identify	Identify
6	Systems engineering	7	2	4	4	1	2	Describe	Describe	Explain	Define
7	Development processes & non-technical aspects (customer requirements, testing, verification, validation, etc.)	6	3	4	4	4	2	Describe	Recognize	Define	Define
8	Project management	2	6	11	11	11	11	Describe	Describe	Recognize	Define
9	Communication – written & verbal; email; presentations	6	3	7	7	7	7	Recognize	Demonstrate	Recognize	Demonstrate
10	Application of theory in real world practice; practical, hands-on skills as gained, e.g., at RRC	2	6	4	1	1	5	Recognize	Explain	Discuss	Identify
11	Balancing business needs vs. technical needs; adjusting technical design to other constraints	2	6	11	11	11	11	Identify	Recognize	Recognize	Recognize
12	Risk assessment, analysis, response; structured analysis methods	5	5	4	2	4	2	Explain	Demonstrate	Describe	Define
13	Cross-disciplinary understanding: knowing enough about other disciplines as they impact one’s work	2	6	1	1	1	6	Demonstrate	Explain	Identify	Recognize
14	Decision-making tools – quantitative and qualitative	1	12	5	5	2	2	Identify	Recall	Apply	Define
15	Interviewing skills	0	14	7	7	7	7	Recognize	Demonstrate		Practice
16	Variability in graduate knowledge, skills, characteristics: ways to reduce variability?	0	14	8	8		1	Locate & Classify	Identify		Identify
17	Professional licensure and regulation	0	14	8	8	8	8	Recognize	Interpret	Recognize	Explain
18	Technical knowledge: representative examples - Software engineering, operating systems, tools that support the technical knowledge, e.g. FPGAs, Linux, kernel drivers,	2	6	1	1	1	5	Locate & Classify	Explain	Explain	Interpret
19	Business functions: design, development, manufacturing, operations, administration, etc.	1	12	11	11	11	11	Recognize	Describe	Describe	Define
20	Societal impact; environmental impact	0	14	9	9	9	9	Recognize	Identify	Recognize	Recognize
21	Engineering ethics; engineering law.	0	14	10	10	10	10	Recognize	Discuss	Recognize	Interpret

4 Conclusions

This paper presents a process and results for conducting an Industry Focus Group forum for curriculum assessment of the Electrical and Computer Engineering (ECE) program at the University of Manitoba. The purpose of the forum was to obtain industry's opinion on the lacking knowledge, skills, and attitudes of new engineering graduates entering the workforce, with the intention that curriculum design and developers use this information to reform the curriculum so that graduates would meet industry's expectations. Twenty-one gap subject areas were identified; they were mapped to the CEAB's graduate attributes for validation; and desired graduate proficiency levels were assigned to CEAB attributes and gap subject areas. Finally, learning outcome statements were formed by assigning Bloom verbs to each subject area based on the selected proficiency levels.

1. We have created an external process (i.e., the IFG forum) to assess the program outcomes of new graduates in the context of the CEAB graduate attributes.

2. We have created a process that is one component in the continuous development and improvement of our program; we will use industry's opinion as feedback, twice a year, for program improvement and curriculum modification.

5 References

[1] Engineers Canada, Canadian Engineering Accreditation Board, 2010 Accreditation Criteria and Procedures, Engineers Canada, Ottawa, CAN, ISSN 1708-8054, 86p. Available: http://www.engineerscanada.ca/e/files/Accreditation_Criteria_Procedures_2010.pdf

[2] B. Frank, "Development of processes and criteria for CEAB graduate attribute assessment", in *CDEN/C2E2 Conference*, Hamilton, Ontario, Canada, 27-29 July 2009. [Online]. Available: http://bmf.ece.queensu.ca/mediawiki/upload/d/d6/CDEN_outcomes_assessment_v1.pdf

[3] "CDIO Syllabus in Topical Form," 2009. [On line]. Available: http://www.cdio.org/benefits-cdio/cdio-syllabus/cdio-syllabus-topical-form.

[4] Washington Accord Rules and Procedures (2007). [Online]. Available: www.washingtonaccord.org.

[5] Graduate Attributes and Professional Competencies. (2009). [Online]. Available: http://www.washingtonaccord.org/IEA-Grad-Attr-Prof-Competencies-v2.pdf

[6] The Engineer of 2020: Visions of Engineering in the New Century, 2004, National Academies Press, Washington, DC.

[7] Educating the Engineer of 2020, 2005, National Academies Press, Washington, DC.

[8] A. S. M. Masud and L. Whitman, "Educating Future Engineers: An Example," in *Proc. of the 2010 International Conference on Industrial Engineering and Operations Management* [Online], Dhaka, Bangladesh, January 9 – 10, 2010. Available: http://www.iieom.org/paper/151%20Abu%20Masud.pdf

[9] R. W. Brennan and R. J. Hugo, "The CDIO Syllabus and Outcomes-Based Assessment: A Case Study Of A Canadian Mechanical Engineering Program," in Proc. Of the 6th International CDIO Conf., Montreal, QC, June 15-18, 2010. [Online]. Available: http://www.cdio.org/files/document/file/T3A_Paper_4.pdf

[10] K. Ferens, "Work in Progress: Are We Teaching The Right Stuff?" in Proc. of FECS'10 - The 2010 International Conference on Frontiers in Education: Computer Science and Computer Engineering, Las Vegas, NV, July 11-15.

[11] G. Cloutier. Email communications.

[12] C. S. Nair and A. Patil, "Re-engineering graduate skills - a case study," *European Journal of Engineering Education*, Vol.34, No.2, May 2009 , pp. 131 – 139.

[13] "ISPE Industry Focus Groups." ISPE, the International Society for Pharmaceutical Engineering. (2011). [Online]. Available: http://www.ispe.org/cs/education_and_training_section/ispe_industry_focus_groups.

Blending Computer Literacy with Computer Science

R. Hatch and J. Somervell
Math and Computer Science Department, The University of Virginia's College at Wise, Wise, VA, U.S.A.

Abstract— *Computer Literacy type courses are almost universally hated by faculty and only slightly less hated by students. They are typically not challenging for anyone involved and are tedious to offer. Canned lectures and rote learning contribute to a sense of boredom with the course. Students are simply engaged in rote memorization and the volume of grading is often overwhelming. We describe a different approach to Computer Literacy topics at a Liberal Arts school that shifts the onus to the students and allows for more interaction and creativity. We report on a pre/post survey of the students of the new course offering and describe faculty response to the course change. This work is intended to show other schools how they might change similar course offerings while still providing certain competencies for their students.*

Keywords: computer literacy, hands-on learning, CS0 education

1. Introduction

Prior to the fall of 2009, the University of Virginia's College at Wise offered a service course (CSC 1100 - Computer Literacy) that provided students with basic knowledge of word processing, spreadsheets, presentation, and database software products; as well as fundamentals of computers and computing. Topics like the Internet, networks, computer hardware, software, and ethical considerations are included. This course provided students the opportunity to meet the college's required proficiencies in Technology Competency, as per the State Council for Higher Education in Virginia's requirements.

Recently, an increasing number of students are arriving at the college as freshmen with most of these skills, and the course was considered "boring" and "not challenging" by the students. Furthermore, the faculty teaching the course complained about the structure and format. The topics were mundane to the Computer Science faculty who offered the course and it was mainly a chore (granted, an easy one) to offer the class and was often used to ease new faculty into teaching. It seemed that neither group (teachers and students) was happy with the course.

In addition, the Department of Mathematics and Computer Science (which offers the Computer Literacy course) was seeking ways to boost enrolment. Meetings with its Advisory Board and internal departmental discussions suggested that the Computer Literacy course could be used to "recruit" some majors, with the idea being that freshmen who take the course and get exposed to some computer science concepts may in fact decide to major in computer science. Some research in the field suggests students may opt to change their majors after taking such courses [5], [3].

In an effort to make the course more interesting for everyone involved, and to possibly increase majors; members of the Department of Mathematics and Computer Science revamped the Computer Literacy course. The remainder of this paper discusses some related works, describes the old and new versions of the course, describes a short pre/post survey used to ascertain effectiveness, and provides discussion.

2. Related Work

Prior work has been accomplished in retooling a CS0 course for several different reasons, ranging from retaining students from CS0 to CS1 courses, to changing the presentation of the course from something of a traditional lecture to a more "hands-on" approach, to change the image of the course, as well as a department to a friendlier, or more interesting course, and one that might be a recruitment tool for students who have preconceived notions of what a Computer Science degree involves.

Purewal notes that there are four main objectives for a CS0 course: 1) a service course that teaches some practical computing knowledge; 2) a course that improves retention and success in other Computer Science courses, one of which is CS1; 3) a course that is designed to recruit a more diverse group of students to Computer Science; and 4) a course that prepares students to analyze and address the current "ethical, social, and legal implications" of increased reliance and "ubiquity" on technology [9]. Previously our CS0 course (named "Computer Literacy") fell primarily under the first of these objectives. With the tweaks made to the course, our department is hoping to touch on each of the four objectives in some capacity. One way to achieve this is perhaps through the introduction of fun or engaging assignments, in which students will try to go beyond the basic requirements and have some fun with assignments [13]. Brady, Cutter, and Schultz view CS0 in a similar light for a liberal arts school – the course is geared toward students who do not intend on majoring in Computer Science; it is a service course [2].

Perhaps the most relevant related work to our project can be found in Cliburn's work [5]. Cliburn notes that the majority of students come to college with a "limited exposure to computer science" at a liberal arts institution. These students, therefore, do not have a true understanding of what a degree in computer science involves, beyond computer literacy (the students can surf the Internet and write papers with some word processor) and exposure to information systems. Cliburn writes about a course that addressed a variety of Computer Science topics and encouraged students to work in pairs on assignments, had the students present on topics in Computer Science, and had debates on ethical issues, where students took sides on topics, such as employers' monitoring the Internet usage of their employees. This CS0 is not a requirement for a Computer Science major, but it is encouraged for those without any programming experience. It is open for other majors, too.

Brady, Cutter, and Schultz suggest a similar strategy to the one outlined by Cliburn; the overriding idea is to have hands-on activities [2]. The class begins with the discussion of the Internet, where everyone is on the same level. Other topics can then be addressed, such as interface design and using technologies to spruce up web pages. Between the lectures and the hands-on activities, topics move to more discussion of the internals of computers.

Purewal offers a different take on how to gain student interest in Computer Science through the offering of a course called "Social Networking." The name alone captures interest, because many students can provide a definition of social networking. From there, other concepts can be discussed, ranging from practical web 2.0 technologies, to mathematical models of social networks, to security and privacy online, and ethical and social issues. One of the interesting findings was an increase in interest in the CS0 course, with just a name change – from "Communications Technologies and the Internet" to "Social Networking" [9]. Malan also discusses introducing some of the geek culture to Harvard's CS50 course by showing LOLcats with linked lists [11].

Mullins, Whitfield, and Conlin discuss using Alice in an introductory programming course [12]. Alice "engages students with 3D animation, a direct manipulation editor for instantiating objects, simple event handling mechanism, (multimedia) capabilities, a simple parallel processing mechanism, and 'syntax-free' IDE." In using Alice for instruction, it can demonstrate some of the basics for algorithms study, such as searching sorting, and even recursive algorithms. It can also show examples of event-driven programming, as well as demonstrate some of the facets of Juman-Computer Interaction (HCI). The assignments for the course vary between submitting a storyboard to requiring methods, functions, or behaviors for library objects. This class has led to an increase in enrollment, as well as fewer withdrawals in this course and the subsequent course. Others (Ludi, Cliburn and Miller, Anewalt) suggest the use of Alice or MIT's Scratch (Malan) as a suitable means to ease students into the discussion of programming or Computer Science concepts [10], [6], [1]. Cliburn and Miller suggest that students might have greater success in understanding concepts through giving assignments that are game-based; the requirements are well-defined and students can determine when they have finished; they prefer assignments that have structure to them [6].

Another issue to consider in the recruitment of students is attempting to recruit a diverse student population. Purewal notes that while interest continues to grow, minorities' and females' interest in the subject does not match interest growth [9]. Part of this, as explained by Carter, can be attributed to the fact that students do not understand or realize what "Computer Science" means or involves [4]. In a survey conducted by Carter, the "vast majority" of respondents did not have an understanding of what the field was or involved [4]. Further, male students majored in Computer Science because of their exposure to video games; female students enrolled in the major to use in conjunction with some other field. Carter suggests reinforcing that Computer Science could be a part of a multidisciplinary program, where the field could be combined with another field [4]. The motivation of such an adjustment would be to dispel that Computer Science is just "programming all day long," as well as bring more females into CS [4]. Some females believe the major to be "too boring" and "overly technical," with no room for creativity [14].

By extension from Carter's research, perhaps faculty could present the students with material that would show them how Computer Science works with other fields, perhaps through the presentation of areas such as HCI. With the discussion of HCI, computer science is present, but the field also interacts with psychology and graphic design, among others, to show diverse fields working together. Women and minorities, according to desJardins and Littman, might be more interested in helping people as the focus of their career [8]. A "Great Insights" of computer science course has been taught at both Rutgers and UMBC, and allows students to acquire an "appreciation for the capabilities and limitations of computing." The intent of the course is to give the students in the other backgrounds (besides CS) some exposure to "scientific principles and mathematical aspects of computer science" [8].

3. Case Study

First we describe the old course in a little more detail. Next we provide an extensive description of the new course and provide a comparison. Then we describe a simple pre/post test to gauge student interest in our majors.

3.1 The Old Course

The old version of the Computer Literacy course followed a highly structured, hands-on approach to office-like products. Textbooks devoted to learning how to use word processors, spreadsheets, presentation software, and databases were used heavily – with the books even showing graphically where and when to click with a mouse. Students completed canned projects from the textbooks, using each type of software during class time, with additional homework assignments for further practice. Prepared slides from the publisher were often used to provide information on the other topics (networks, hardware, software, etc), with short quizzes or multiple choice tests (again provided by the publisher) to test the students. Often the test questions from the test banks would be obscure usage questions:

To center text on the screen use the — shortcut.
a. CTRL+L
b. CTRL+E
c. CTRL+R
d. CTRL+1

The course was universally dreaded by the faculty because it was boring, yet work intensive (grading); and students complained about the course. It offered no challenge to either the instructors or the students.

3.2 The New Course

Any changes to the course needed to be done internally, while still meeting the required technology competencies of the college. The new course we created still covers the same types of material as the old course, but in a different way. To overcome the "boring" approach of the old course, the new course relies more heavily on creativity and student oriented work. Instead of using a specific text to teach the office concepts, we adopted a hands-on, active-learning approach in which the students use the products to solve some other problem. For example, to help students learn word processing techniques we might assign a short (2-3 page) research or opinion paper. The student must use some word processor (we do not specify a specific piece of software) to create their report. We also take the opportunity to introduce other useful concepts like creating PDF versions of documents and creating zip archives.

In addition to having the students use the software in a meaningful way, we also make sure the students are exposed to alternative software packages. For example, we ask the students to use two different software packages to solve the problems, then write a report on the students' preferences. Our students have chosen packages like Microsoft Office, Open Office, and Google Docs, among others.

We shift the onus of presenting various technology topics to the students. We ask them to work in pairs or teams of three to present certain topics to the rest of the class. Example topics might include the history of the Internet, social networking, e-commerce, and identity theft. The student teams research their topics and present their findings to the class. This also reinforces their office skills because they must write a report and create a presentation. An added bonus of this approach is the exposure to in-class presentations and a chance to work on oral communication skills.

We also added a hands-on hardware component. Students are divided into teams of 4 or 5 and provided with all of the components of a desktop computer: the chassis or tower, a motherboard, a processor and fan, memory modules, power supply, hard drive, and optical drive (DVD). The students then assemble the computer and install a new operating system (we chose Ubuntu). The students seem to enjoy this aspect of the class and many have noted a boost in confidence with working with hardware.

To introduce some important ethical considerations we have the students research and defend, either for or against, some controversial topic. A specific example is the idea of Net Neutrality. Students who are supposed to argue for Net Neutrality would present ideas about freedom of expression, equal access, privacy, etc. Students arguing against Net Neutrality would present ideas about capital investment, quality of service guarantees, law enforcement, etc. Topics are discussed in class using a debate style approach. The students seem interested and motivated, and usually have strong opinions about these topics.

A final change to the course is the inclusion of basic computer science concepts. Traditionally the course provided no introduction to computer science. We wanted to expose the students to the basics of problem solving using a computer. The last few weeks of the course involve an introduction to problem solving and algorithmic approaches through simple programming. We use the Alice programming language and environment [7].

The goal is to provide some idea of what computer science is about and get the students excited about creating their own programs. Typical assignments include creating environments, and simple animations. The focus is on identifying the steps needed to solve a problem and applying basic programming concepts, like sequencing, selection, and repetition, to solve the problems.

3.3 Comparison

The new course is significantly different from the old one. The entire presentation approach now relies on heavy student input as opposed to a passive, lecture style. Students have ample opportunities for creativity throughout the course, instead of working on canned projects. We shortened the focus on office skills and included materials on ethical considerations, building computers, and programming. We also encouraged the students to attend a local technology symposium by offering extra credit for attending sessions and writing short reports on those sessions. We feel the new course more accurately matches what a "Computer Literacy" course should be.

In addition, the new course is well-received by the faculty. The new course is more flexible and allows each instructor to choose the specific topics to be covered. This means that the course is no longer "boring" for the instructors. While we do not have quantitative data, we also expect that students find the new course more interesting, useful, and challenging than the older course. We feel the new course is an improvement over the old and we plan on using the new model in future offerings.

3.4 Pre/Post Test

To ascertain whether the new course changed students' opinions about majoring in Computer Science we performed a simple pre/post survey. On the first day of classes we had the students complete a survey asking about their opinions of computing and majoring in computer science. Questions addressed student opinion on working with computers and interest in Computer Science, Software Engineering, and Management Information Systems. We used Likert-type questions with a 5-point scale. In addition, we asked open ended questions about the amount of computer usage and for what purposes. The following list provides the questions from the pre-test:

1) How often do you use computers (hours per day)?
   0 - 2 2 - 4 4 - 6 more than 6
2) For what purpose do you use computers most?
   work school socializing entertainment other(specify)
3) How do you feel about working in technology related fields?
4) I find computers boring.
   (strongly agree to strongly disagree for scale)
5) I think working with computers is more of a "geek thing."
6) I enjoy working with computers.
7) I feel that learning about computing is important.
8) How interested are you in Computer Science?
   (not interested to very interested for scale)
9) How interested are you in Software Engineering?
10) How interested are you in Management Information Systems?

The post-test differed only slightly by the addition of the following question:

> Please tell us how this course has changed your opinions about Computer Science, Software Engineering, and Management Information Systems.

4. Findings

4.1 Pre-Test

Twenty-four students filled out the pre-test questionnaire. The average age was 19.4 with a median age of 19. Fourteen students were male, 8 were female (2 did not specify). All students used a computer on a daily basis with 35% using a computer 4-6 hours per day. Forty-five percent of the students specified work or school as the main reason for computer usage. Seventy-five percent of the students had a positive attitude towards working with computers.

The average response to the statement, "I find computers boring" was "disagree" with no major differences between males and females. Similarly students disagreed with the statement, "I think working with computers is more of a geek thing" with no differences across gender. Gender played no roll in opinions of working with computers and learning about computing (with average responses on both questions as "agree").

In terms our specific majors, Computer Science, Software Engineering, and Management Information Systems; students mainly "did not know" what their interest level was.

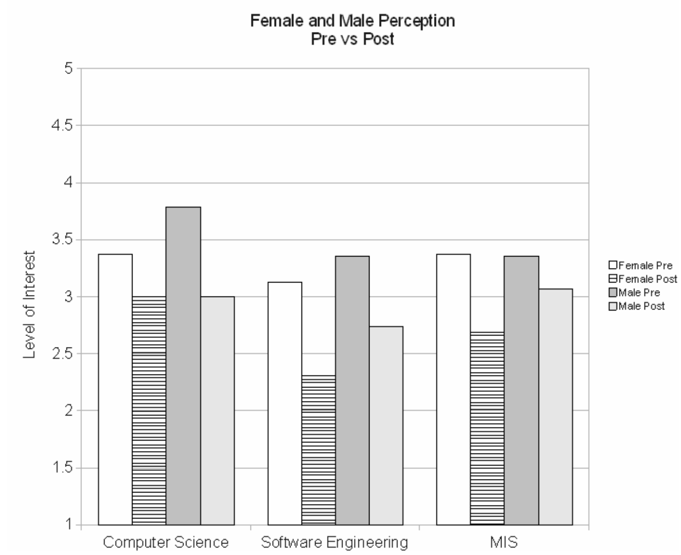


Fig. 1: Female and male perception changes from pre to post test.

4.2 Post-Test

Twenty-nine students filled out the post-test questionnaire. The average age was 19.5 with a median age of 19. Fifteen students were male, 13 were female (one did not specify). Computer usage was greater than reported on the pre-test with 67% using a computer more than 4 hours per day. Again the major usage for the computer was school or work with 64% indicating those areas. Interestingly, only 43% of the students reported a positive attitude towards working with computers after taking the course. This is down from the 75% reported in the pre-test.

Responses to the questions probing opinion of computing were comparable to the pre-test findings. Specifically the average student response to the statement, "I find computers boring" was "disagree." Males actually had a lower average of "strongly disagree" than the females. This was a change from the pre-test. The response to the statement, "I think working with computers is more of a geek thing" remained the same, "disagree," with no differences across gender. The response to the statement, "I enjoy working with computers" was "agree," and again no change from the pre-test. One interesting finding is that the average responses about interest level for the majors went down. It is not a statistically significant difference, but interest in all three majors dropped from the pre-test to the post-test (dropping from "do not know" to "slightly disinterested." The interest level for females dropped more than for males. Please see figure 1.

The extra question on the post-test was intended to glean the students' opinions about the class and how it changed their outlook on computing. To analyze the responses, they were categorized into positive and negative comments. Positive comments mentioned something good about the course or a positive outlook towards computing. Negative comments typically had a complaint or a negative outlook towards computing. Nineteen of 21 responses were positive (seven students did not provide a comment). Some example positive comments include:

> "[It] helped me understand how computers are used in today's society"
> "[I] learned a lot about programming and building a computer"
> "Very informative and helpful, I can say I have put a computer together"

The two negative comments we received were complaints:

> "Get rid of Alice and let her go to Wonderland"
> "[I] like the class for knowledge, but busy work was frustrating"

4.3 Discussion

Comparison of pre- and post-test data indicates that students' opinions of Computer Science, Software Engineering, and Management Information Systems changed slightly in the negative direction. It is believed that the students gained a clearer understanding of what those topics involve and gave more accurate answers. Instead of simply choosing the vague "I don't know," they chose the more accurate answer of "disinterested." This is disheartening because a goal of the change was to try to increase majors. We do not currently have data to determine if the new course has generated more majors over the prior course.

The high rate of positive comments from the students indicate that they typically enjoy the new course. That knowledge is comforting and coincides with our goal of making the course more interesting and student centered. Unfortunately, that is the extent of what we learned from the students. What we learned from the faculty is much more interesting and salient.

Faculty Response

Through interviews with the faculty who taught the course, it was determined that the new course offering was "better." First, the course is highly individualized to the professor who teaches it. This requires some preparation but the benefit is that the topics, tone, and outcome of the course will match an individual instructor's desires. For example, an instructor interested in computer security could focus the course on security threats, steer ethical discussions towards password sharing or identity theft, and include social engineering techniques as examples.

Second, the course requires students to take charge of a significant portion of the course. Either individual or team presentations are required on current topics in computing. Example topics might include security and threats, digital copyright, personal data access at work (Facebook while at work), Internet technologies, etc. The students are responsible for researching their assigned topic, creating a presentation, giving a lecture (a whole class period), and leading discussion. This allows the instructor to focus on the content and engage the students rather than on planning lectures. In sum, instead of rote learning, where students memorized definitions for exams; the students were charged with providing content to the course, making them active learners.

Third, the inclusion of programming topics has increased faculty buy-in. The course is no longer just "teaching how to use Microsoft." It is now a chance to introduce some fundamental problem solving skills through algorithm design and implementation. The Alice programming environment is fun and supports creativity. Instructors are allowed to choose any projects they wish, with the caveat that the students will need help and training in how to solve them.

5. Conclusion and Future Work

The transition from a traditional lecture style course with canned projects to a more interactive, student-centered approach has been a positive change. The most salient change is the increased satisfaction reported by the faculty.

The pre/post test did not show any significant changes in attitude by the students. Most of the students came to the class with no intention of majoring in Computer Science, Software Engineering, or Management Information Systems, nor having a clear understanding as to what each field involves. The course did not seem to change those intentions. However, the course seemed to be well received by the students and the new course methodology will continue.

In terms of future endeavors and continued evaluation, more detailed studies of faculty response are planned. Specifically, a more quantitative assessment of faculty opinion are required to more fully ascertain the strengths and weaknesses of the new approach. Also, new ideas will be considered to attempt an increase in majors who come from this course and quantitative data will be analyzed to determine matriculation rates.

References

[1] K. Anewalt. Making cs0 fun: An active learning approach using toys, games, and alice. *Journal of Computing Sciences in Colleges*, 23(3):98–105, Jan. 2008.

[2] A. Brady, P. Cutter, and K. Schultz. Benefits of a cs0 course in liberal arts colleges. *Journal of Computing Sciences in Colleges*, 20(1):90–97, Oct. 2004.

[3] R. Bruce and S. Reiser. Introduction to technology for general education. In *Proceedings of the 43rd Annual ACM Southeast Conference*, pages 274–279, Kennesaw, GA, USA, Mar. 2005.

[4] L. Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, pages 27–31, Houston, TX, USA, Mar. 2006.

[5] D. C. Cliburn. A cs0 course for the liberal arts. In *Proceedings of the Technical Symposium on Computer Science Education (SIGCSE 2006)*, pages 77–81, Houston, Texas, USA, Mar. 2006.

[6] D. C. Cliburn and S. M. Miller. Games, stories, or something more traditional: The types of assignments college students prefer. In *Proceedings of the 39th ACM Technical Symposium on Computer Science Education*, pages 138–142, Portland, OR, USA, Feb. 2008.

[7] W. P. Dann, S. Cooper, and R. Pausch. *Learning to Program with Alice*. Pearson,Prentice Hall, 2009.

[8] M. desJardins and M. Littman. Broadening student enthusiasm for computer science with a great insights course. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pages 157–161, Milwaukee, WI, USA, Mar. 2010.

[9] T. S. P. Jr. Social networking: The new computer fluency? In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pages 112–116, Milwaukee, WI, USA, Mar. 2010.

[10] S. Ludi. Active-learning activities that introduce students to software engineering fundamentals. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 128–132, Monte de Caparica, Portugal, June 2005.

[11] D. J. Malan. Reinventing cs50. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pages 152–156, Milwaukee, WI, USA, Mar. 2010.

[12] P. Mullins, D. Whitfield, and M. Conlin. Using alice 2.0 as a first language. *Journal of Computing Sciences in Colleges*, 24(3):136–143, Jan. 2009.

[13] P. M. Mullins and M. Conlin. Engaging students in programming fundamentals using alice 2.0. In *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education*, pages 81–87, Cincinnati, OH, USA, Oct. 2008.

[14] R. H. Sloan and P. Troy. Cs 0.5: A better approach to introductory computer science for majors. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, pages 271–275, Portland, OR, USA, Mar. 2008.

K-12 Computer Education Deficiencies in Nevada

Stephanie Shreck
Shahram Latifi
University of Nevada, Las Vegas
Department of Electrical and Computer Engineering
University of Nevada, Las Vegas
4505 Maryland Parkway
Las Vegas, NV 89154-4026

Abstract

A study done in 2009 found that pre-Advanced Placement (AP) Computer Science (CS) classes for high school students are declining. This, along with the decreased enrollment in Computer related studies at a collegiate level, paint a grim picture for the future of the computer education in the US. Indicated contributors to this decline include inadequate teaching resources and expertise, as well as the perceived outsourcing and societal implications associated with computer related career choices. In an attempt to overcome these obstacles, organizations like the National Science Foundation (NSF) have funded programs targeted at developing primary and secondary computer education courses. The current research is a study that analyzes Nevada's primary and secondary education needs, discusses the issues related to computer education, and evaluates several NSF funded programs. Proposed are preliminary steps that need to be taken by the Nevada System of Higher Education (NSHE) to help correct the direction of computer education in Nevada.

Keyword: CRT, HSPE, Math, Science, Testers to Techies, Underrepresented Group.

1. Introduction

As technology continues to advance, the role of computing is becoming more imperative to all aspects of the workforce. From digitizing medical and legal documents to utilizing social networking tools as a strategic business tool, to be successful in almost any career, computing acumen is vital.

Those who are successful in business have learned how to maximize the use of computing technologies to increase business efficiency, public exposure, and product development. Products like smart phones and iPads provide society with the world's information at their fingertips. By allowing primary and secondary education students an opportunity to develop computing skills at early ages, their ability to succeed in post high school career choices is greatly increased.

In Nevada, the service industry, which accounts for 41% of Nevada earnings, is being revolutionized by technological advancements [1]. The technology that has been added to the gaming industry allows for better security in the high traffic casinos, more reliable money transactions, and web services that entice guests with convenience. In order for Nevada to maintain its highly regarded vacation destination status, the incoming work force will need to be prepared to provide services that advance as fast as commercial technical fads do.

The large impact of the service industry on Nevada earnings makes the state highly dependent on the national and world economic climate. Tourists only take part in the luxuries that Nevada's service industry provides them, when they are economically able to do so. In other words Nevada's prosperity relies on the prosperity of other states, countries, and their tourists. This type of interdependence can be seen in Nevada's unemployment rate of 13.7% (based on Google's October 2010 numbers), the highest of any state and almost 5 points higher than the national average.

Providing students with a background in computing can help increase Nevada's scientific community and entice more youth to continue in science and engineering jobs. This provides Nevadans with the skills to seek other job opportunities in times when the service industry is limited.

2. Statistics in Nevada Education

In providing Nevada youth opportunities to develop and practice computational competencies, there are some key challenges that will need to be addressed. Nevada school districts are made up of a diverse group of students. The pie chart in Figure 1 shows the distribution for students in pre-kindergarten through 12th grade across the state.

The mix of students from various cultural backgrounds produces challenges when developing course curriculum that will engage all students. This engagement is necessary to encourage students in continuing the development of these skills through post-high school education opportunities.

Nevada has also seen a decline in math and science scores as students' progress through primary and secondary grades. In the minority groups this decline is even more dramatic. Table 1 shows the Criterion Referenced Test (CRT) and High School Proficiency Exam (HSPE) scores by subgroup and grade for Nevada's largest county - Clark.

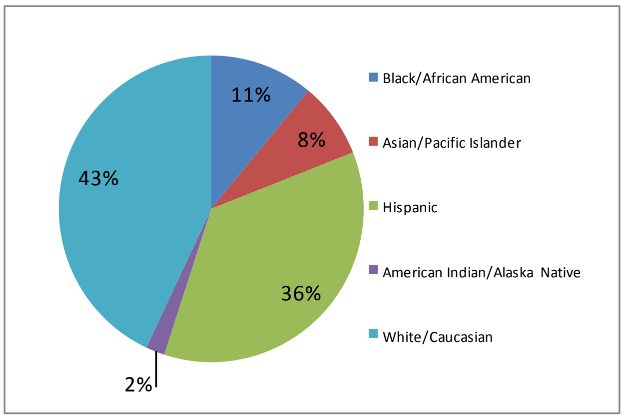


Figure 1 - Nevada PreK-12 Student Ethnicity Distribution

While the declining scores are worrisome in themselves, the key drops between 7th and 8th grade and then again from 8th to 10th are of particular interest. These scores juxtaposition with Nevada's 47.3% high school graduation rate (20 points lower than the national average) indicates that the timing of when the curriculum is first introduced to students and how it is sustained through subsequent grades will be extremely important to any program's success [6].

Based on the Nevada demographics and test scores, providing middle school students with programs that can intrigue them and high school courses that can expand the knowledge will be important to the success of any computer education curriculum.

3. Issues in Computational Education

Before determining possible approaches, it is important to understand some of the issues that hinder computer technology education in society. Nagel identifies several reasons by teachers as to why Computer Science courses are declining in the US [3]. The top three observations listed in the article are:

1. Rapidly Changing Technology
2. Lack of Staff Support of Interests
3. Lack of Curriculum Resources

To address these issues it is important to develop an infrastructure that can be fluid with the changing technologies. By keeping curriculum up to date with the current technologies, the interest of students in the program will be maintained.

Providing teachers and support staff the capabilities to support the curriculum is vital to the success of a program. Computer Science skills are often a difficult subject matter for both students and teachers. Making sure teachers are comfortable in the area allows students to be more successful in post high school education.

Funding for primary and secondary education usually goes to one of two purposes [3], state required activities and student high interest activities. Computer Science is not currently a required area for K-12 students, and therefore it is imperative to have students interested in engineering and computer science to maintain funding.

Several other problems associated with the decline in computer science include the stereotypical image of a computer scientist or engineer, outsourcing of technical jobs, and programming deficiencies once in post high school education.

The stereotypical image that often comes to mind when students start thinking about computer science and engineering careers is not appealing to youth. Computer scientists are often portrayed as anti-social men that work in isolation. This image is the antithesis of primary and secondary education objectives that focus on promoting participation in group activities and the importance of teamwork. Reversing this image in the minds of students and portraying a more accurate view of computer science and engineering will be important to attract secondary education students.

The image and attitudes that are perpetuated in the first few years of computer and engineering courses also limit minority involvement. In a study performed in Teague and Roe, they found "women often perform well academically, yet perceive the programming environment as inhospitable, lacking social meaning and interaction which is incongruent with the real world" (p. 148) [4]. Showing the social purpose of computing careers and how life has changed throughout the years due to the advances

Table 1 - Percent of Clark County Students in 2009 Meeting or Exceeding Standards in Math (M) and Science (S) Scores by Subgroup (Criterion Referenced Test (CRT) used in 3rd-8th grade and High School Proficiency Exam (HSPE) used in 10th grade) [2]

Subgroup	3rd	4th	5th	6th	7th	8th	10th	5th	8th	10th
	M	M	M	M	M	M	M	S	S	S
State Average	61	64	62	67	63	55	49	56	61	62
Female	59	64	63	67	64	53	45	53	59	55
Male	62	64	61	64	60	53	48	55	60	62
Black/African American	43	49	46	48	45	36	28	37	42	39
Asian/Pacific Islander	74	79	77	82	80	72	64	67	75	71
Hispanic	54	58	55	57	53	41	33	43	47	45
White/Caucasian	73	75	74	78	74	68	63	72	76	76
Students qualifying for free/reduced lunch	51	55	52	55	50	40	33	41	45	44
Students with disabilities	34	34	27	24	19	13	9	27	23	18
Students with limited English Proficiencies	49	48	37	33	26	16	12	21	16	12

made in this field, should be another goal in getting students excited about computer science.

Outsourcing is an issue that limits the attractiveness of computer science careers. In a success driven society like that of the US, it is important to students that they select career paths that have high potential growth, both financially and personally. As companies continue to move their IT divisions to countries like China and India, the perceived job potential and value of the computer science skills that are needed to enter into these career paths decreases substantially.

In order to rectify outsourcing, companies need rich pools of talented US students who are able to fill the positions as they are needed. With the declining enrollment in computer related fields, it is difficult to know whether this image can be altered in the near future. However by educating students in the versatility that the computer science skills provide, steps can be made to bring these technical jobs back into the US.

In studies performed by Computer and Engineering departments on why students drop out of the discipline, it was determined that the first programming course taken has a large impact on the choice of students to stay or leave. Deficiencies in an understanding of simple computer skills make programming courses difficult for students. The anxieties associated with these deficiencies, as well as the unfamiliar structure of programming, make it hard for students to become comfortable with the techniques. Providing students computer science competencies and exposure to programming structure before college, will allow students to become more comfortable with the technologies and concepts. This should help reduce the anxieties, enabling students to continue their enrollment in computer and engineering degree programs. [10]

4. Programs in Computer Education

Several studies and programs have been developed to create a curriculum in computer education for K-12th grade students. To move quickly to rectify the declining state of Nevada's computer education, it is important when developing curriculum to utilize lessons learned.

"Exploring Computer Science" (ECS), lead by Dr. Joanna Goode, is a very successful computer science curriculum that was developed for the Los Angeles Unified School District (LAUSD) [8]. The program was developed to address the disproportionately low Latino and African American students who are pursuing advanced degrees in computer science. The foundation set forth in the ECS program currently being used in the LAUSD and being expanded to San Diego, can be leveraged and customized to produce curriculum and summer workshops that can meet the needs of Nevada students.

The ECS curriculum provides high school students a year long course with interactive units covering a variety of computer science areas. Nevada's high school graduation rate, as mentioned previously, is more than 20 points lower than the national average. Due to this statistic, Nevada will require that the interest of students be peaked prior to getting to high school and then sustained through efforts like ECS.

Other programs like "MOBILIZE", lead by Dr. Debra Estrin from UCLA, allow students to pursue scientific studies through the use of participatory sensing technologies like cell phones [7]. Linking science with something as widely used as cell phones, provides students a look at the power of computational technologies in easing daily life activities. Because most Nevada jobs do not require the type of continued education that ECS gears students for, activities like MOBILIZE will be important. This curriculum will show students the advantages of having foundational computing competencies, how they can enhance daily activities, and support general workforce jobs.

Another program, “Testers to Techies” (TtT), lead by Dr. Janice Cuny from Georgia Institute of Technology, looks at methods to break through cultural barriers that exist [9]. By leveraging the interest of teenage boy in video games and offering apprenticeships as game testers, TtT is directed at engaging low-income African American youth in computing. Video games are universally enjoyed by youth and developing this interest can provide the excitement needed for students to continue learning about computational sciences.

5. Moving Forward in Nevada

In order to start making progress in computer education, representatives from the Education and Engineering Departments within the Nevada System of Higher Education (NSHE) need to combine forces with selected schools in the Clark County. This alliance should hold workshops and collect surveys from teachers statewide to refine its understanding of the Nevada specific barriers associated with computing education. By utilizing the foundation set forth in programs like “Exploring Computer Science,” the alliance can create curriculum and summer workshops that can meet specific Nevada needs.

The alliance would need to perform research in designing curriculum that can leverage the interests of students in current technology. Developing course materials related to cell phone applications, Facebook, video games, and other technologies, can provide students fun, relevant, hands-on activities. The goal is to provide students with computing competencies that can assist them in daily life and the industries that dominate Nevada culture. In addition, they will be introduced to careers in math, science, and engineering.

While the standardized test scores speak for themselves, it will be important to provide more background by performing workshops with Nevada math and science teachers. These workshops will help to understand what the challenges of the teachers are on a daily basis from a student and resource perspective. Developing a relationship with these teaching professionals will be an essential aspect of any preliminary work. The initial work will be to understand needs of the teachers, with subsequent work introducing several curriculum options to perform test trials and elicit feedback.

These workshops would also include getting feedback from community college and university professors in the areas of math, science, and engineering. Working with this group of teaching professionals would provide insight into what skills students need to be successful in post high school academics. By incorporating feedback from this group, there is a top down view on what trends are seen at this collegiate level. This will also ensure that any high school curriculum that is developed will be adequate in preparing the students for the challenges they may face at this level.

Student surveys will also be a necessary part of providing context to test scores and will complete a 360-degree view of the issues. This allows the alliance to understand what students of different ages are interested in, as well as potential career paths. With a better understanding of what excites students, the curriculum can be geared to align with those interests.

Given Nevada’s large Hispanic population, focusing activities that are directed for this underrepresented group are also necessary. Providing fundamental computing courses in Spanish for bilingual or English deficient students may provide a path to overcoming the language barrier that is making success in math and science fields difficult.

Leveraging groups like the Girl Scouts of America can also be advantageous in getting females, another underrepresented group, interested in computing fields. Doing summer workshops with a specialized group like this could make careers in math, science, and engineering more accessible to young women.

6. Conclusion

In order to provide the educational opportunities for Nevada students to develop and practice computational competencies, there first needs to be an infrastructure in place to support the efforts. A planning phase to create an alliance with the NSHE and selected schools in the Clark County is the first step. The alliance will need to analyze the problem from the 360-degree perspective of: primary and secondary students, primary and secondary educators, and community college and university professors.

Following the planning phase, proven computing curriculum needs to be customized for Nevada’s specific needs. Finally, small trials of the program need to be performed to determine its efficacy.

Nevada's declining test scores and low high school graduation rate are causes for concern. Linking the interests of students in today's technology trends, like ipads, cell phones, and Facebook, with computer education could be one effective method to help reverse this decline.

7. Acknowledgment

This work was supported in part by the Nevada EPSCoR Programs, and funded by NSF "Grant # NSF-NV 0814372" ,and in part by University of Nevada, Las Vegas Graduate College.

8. References

[1] Nevada – Industry, City-Data.com, http://www.city-data.com/states/Nevada-Industry.html.

[2] Nevada School District Statistics, EducationBug, http://nevada.educationbug.org/school-districts/.

[3] Nagel, David. "Computer Science Courses on the Decline." The Journal: Transforming Education through Technology 04 August 2009. http://thejournal.com/articles/2009/08/04/computer-science-courses-on-the-decline.aspx.

[4] Teague, D., & Roe, R., "Collaborative learning – Towards a solution for novice programmers." Proceedings of the Tenth Conference on Australasian Computing Education - Volume 78, 147-153, 2008.

[5] Clark County School District CRT and HSPE Results, http://www.greatschools.org/cgi-bin/nv/district-profile/2.

[6] Gibbons, Patrick. "Graduation time: Hold the celebration", Nevada Policy Research Institute, 9 June 2010. http://www.npri.org/publications/graduation-time-hold-the-celebration.

[7] Shilton, Katie. "Four Billion Little Brothers? Privacy, Mobile phones, and ubiquitous data collection." Communications of the ACM 52.11 (2009): 48-52. http://queue.acm.org/detail.cfm?id=1597790.

[8] Goode, Joanna. "Exploring Computer Science (ECS)." http://www.exploringcs.org/.

[9] Cuny, Janice. "Collaborative - BPC-DP: Testers to Techies: Culturally Aware and Authentic Computing Education through Game Testing." http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0837650.

[10] Lester, C.Y.; , "Growing the Pipeline: Restructuring an Introductory Computer Programming Course," *Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on* , vol., no., pp.23-28, 11-13 July 2007. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4276352&isnumber=4276339.

Design Business Use Cases for a Data Mining Course Based on an Enterprise Level Database

Ying Xie, Young-Seok Kim
CSIS Department, Kennesaw State University, Kennesaw, GA, USA
yxie2@kennesaw.edu, ykim35@students.kennesaw.edu

Abstract - *This paper proposes four comprehensive use cases that enable students to conduct data mining in a comprehensive business data environment with the goal of meeting specified business requirements. The use cases are designed based on the following frequently used data mining techniques: association mining, classification, clustering, and time series prediction.*

Keywords: Data Mining, Association, Classification, Clustering, Time Series, AdventureWorks, Use Cases

1 Introduction

Data Mining is one of the key members in the family of the Business Intelligence techniques. It is about discovering previously unknown and potential useful patterns from data. Given the fact that ever increasing amounts of data have been consistently accumulated in various businesses, the capability of making intelligent use of the data may largely enhance the competitiveness of the business. Typical business applications of data mining include but not limited to recommendation generation, customer segmentation, targeted ads, churn analysis, risk management, anomaly detection, and forecasting [3].

Information Systems (IS) is an academic discipline that equips students with knowledge in technology-enabled business development [1]. Since data is ubiquitous and critical in business world, data management and analysis is an essential skill that an IS major needs to grasp. Going beyond a typical course on database management systems, the IS program at Kennesaw State University offers a data mining course to IS majors with the goal of making students more competitive and marketable in the area of business intelligence.

Like other information technology courses offered to IS majors, the data mining course should be taught from a business-oriented perspective. Therefore, instead of focusing on the design and analysis of data mining algorithms, emphasis should be given on the business applications of different data mining techniques. As a critical learning outcome of an IS Data Mining course, students should be able to apply different data mining techniques to conduct intelligent data analysis on an enterprise level database or data warehouse. This learning outcome requires students be provided with multiple business use cases for data mining based on an enterprise level database or data warehouse. However, it is quite typical that a textbook uses a single table/spreadsheet, or a simple database composted of a couple of tables to illustrate each data mining technique [2, 4, 5]. In other words, a comprehensive business data environment is typically lacking in a data mining textbook.

The book “Data mining with Microsoft SQL Server 2008” [3] is of exception to some extends. It uses two enterprise level sample databases AdventureWorks and AdventureWorksDW to illustrate some of the data mining techniques. Both AdventureWorks and AdventureWorksDW are provided by Microsoft as sample databases designed for a fictitious business AdventureWorks. AdventureWorks is an OLTP database that supports the daily business of AdventureWorks; while AdventureWorksDW is a data warehouse that supports business analyses. These two databases provide a good business data environment for students to apply learned data mining techniques. However, instead of presenting some comprehensive business scenarios for applying data mining techniques to a comprehensive database, this book directly uses some existing views pre-generated in the AdventureWorksDW as media for covering related data mining techniques. Essentially, this approach is not so much different from using a single table or a spreadsheet. In order to equip students with practical capabilities of conducting data mining techniques on an enterprise level database, we propose a set of business use cases that requires students federate proper data from multiple tables in a comprehensive database and then apply proper mining techniques to address a given business need. Considering the undergraduate students who take the data mining course may not have prerequisite on data warehouse, we design the use cases based on the sample OLTP database AdventureWorks. The designed uses cases cover the major data mining techniques including association mining, clustering analysis, classification, and time series analysis. We will describe the usage of these use cases in section 2, and then provide detailed description for each use case in section 3-6. The conclusion of this paper is given in section 7.

2 Coursework Based on the Proposed Use Cases

Each use case starts with a brief description of a business need. Then students are required to locate the relevant data in order to address the business need. This requirement actually provides students with opportunities where they apply knowledge on database schema learned in a prerequisite database course to the analysis of a comprehensive database. Next, students need to identify the proper mining technique and then design a proper target table or view for the mining. Subsequently, it requires students to populate the target table or view by federating data from relevant tables. There are two options for federating relevant data. The first option is to write complex SQL Statements for the purpose. This option allows students to practice and deepen their knowledge on SQL. The second option is to use integration service provided by a DBMS such as the SQL Server Integration Service (SSIS) provided by Microsoft. This option provides students opportunity to learn advanced tools for federating and transforming data by building data flow pipelines. Finally, the identified data mining techniques are applied and mining results are analyzed.

A use case can be discussed in lecture when a particular data mining technique is taught. However, it is ideal that the complete set of the proposed use cases are used in a comprehensive course project. In this project, students are provided with a set of business needs and required to go through all the steps mentioned above in order to satisfy each business need. Given different study progresses of the course, this project can be tailored in different ways ranging from providing hints on how to locate relevant tables for each need to giving partial solution on data federation. Each use case can also be used separately in a hands–on assignment right after the corresponding data mining technique is discussed.

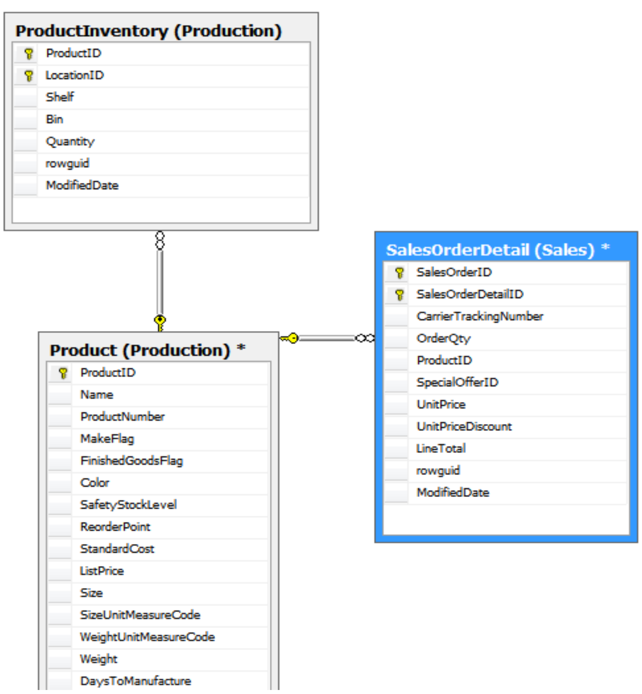


Figure 1. Schema for tables needed for association mining

3 The Proposed Use Case on Association Mining

The business need for this use case is to generate a mining model that is able to recommend products with high inventory when a customer purchases an associated product. Business analysis based on this need requires product information, product inventory information, and product sales order information. As the first step to address this business need, we identify the following relevant tables from Production and Purchasing Department as shown in Figure 1.

Next, we identify the proper data mining technique for this analysis as well as the target data view for the mining. Given that association mining can be used to build recommendation engine based on frequent bundles of products in same transactions, it is identified as the core data mining technique for this analysis. The challenging part of this analysis is that a recommended product should have high inventory. Therefore, we require each transaction in the target data view must contain at least one product with high inventory (inventory value is greater than a threshold). A target data view for association mining is shown as in Table 1. Afterwards, we need to populate this target data view by retrieving relevant data from the three tables identified in the first step. The SQL statement for the retrieval is shown in Figure 2.

OrderID	Name	Inventory
1	Adjustable Race	1085
2	Thin-Jam Hex Nut 9	1103
2	Thin-Jam Hex Nut 10	1084
3	Seat Post	780
4	Headset Ball Bearings	1322
5	HL Road Rim	615
6	Touring Rim	1364
7	HL Crankarm	797
7	LL Crankarm	593
7	ML Crankarm	439

Table 1. Target data view for association mining

```sql
SELECT  sod.SalesOrderID,
        pp.Name,
        pi.Quantity AS Inventory
FROM Production.Product pp
JOIN Production.ProductInventory pi
ON pp.ProductID = pi.ProductID
JOIN Sales.SalesOrderDetail sod
ON pp.ProductID = sod.ProductID
ORDER BY 1, 3 DESC
```

Figure 2. SQL statement for populating the target data view

4 The Proposed Use Case on Classification

The business need for this use case is to generate a mining model that is able to predict the order status for a new purchase order. By studying the whole database, one may realize that the order status may be affected by purchasing information, vendor situation, and product characteristics. Therefore, we identify the relevant tables for this analysis as shown in the Figure 3.

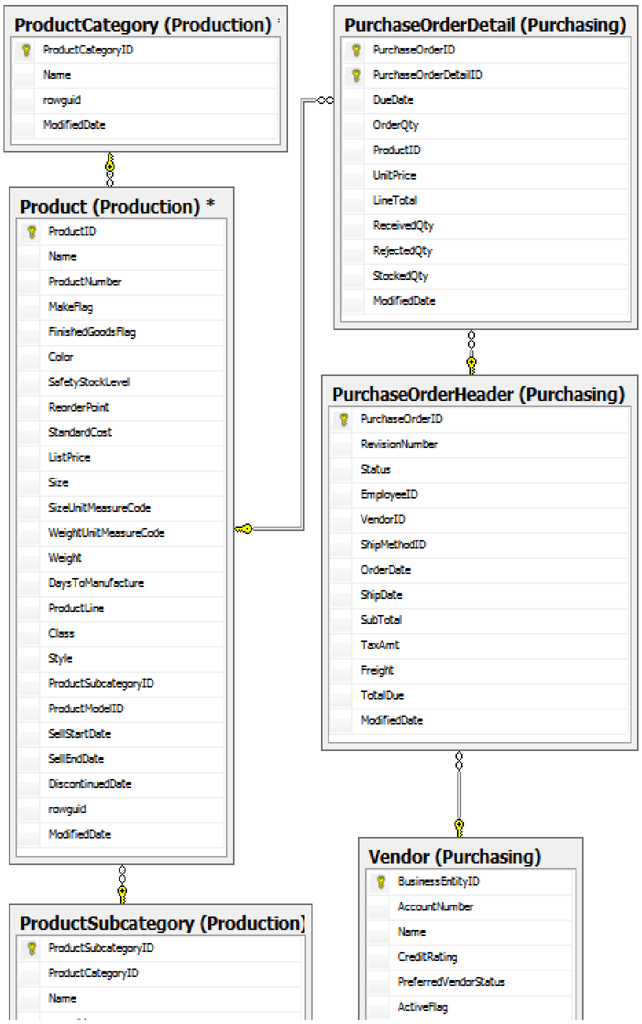


Figure 3. Schema for tables needed for classification

Then, a classification algorithm, such as Decision Tree or Naïve Bayies, will be used to build a classifier that reveals how input attributes on purchasing, vendor, and product affects order status. Given that the relationship between PurchaseOrder and Product are one to many, the target mining structure for classification should contain a nested table column on products for each purchase order, as shown in Table 2. The population of the identified mining structure needs to include two SQL statements as shown in Figure 4. The second SQL statement in Figure 4 is used to populate the nested table column of the identified mining structure shown in Table 2.

5 The Proposed Use Case on Clustering

The business need for this use case is to generate a mining model that is able to segment customers based on customer types (individual or business), total amount of money spent on purchasing, and the set of products ever purchased. The involved information includes customer, sales order, and product. Therefore, we identify the schema of relevant tables for this analysis as shown in Figure 5.

```
// Purchasing & Vendor Information
SELECT poh.PurchaseOrderID,
       poh.SubTotal,
       poh.TaxAmt,
       poh.Freight,
       poh.TotalDue,
       poh.ShipMethodID,
       poh.Status AS OrderStatus
       pv.Name AS VendorName,
       pv.CreditRating,
       pv.PreferredVendorStatus,
       pv.ActiveFlag
FROM Purchasing.PurchaseOrderHeader poh
JOIN Purchasing.Vendor pv
ON pv.BusinessEntityID = poh.VendorID
ORDER BY 1

//Product Information
SELECT  ppod.PurchaseOrderID,
        pp.Name AS Product,
        pc.Name AS Category,
        PSC.Name AS SubCategory,
        pp.Color,
        pp.ListPrice,
        pp.Size,
        pp.SizeUnitMeasureCode,
        pp.Weight,
        pp.WeightUnitMeasureCode,
        pp.Class,
        pp.Style
FROM Purchasing.PurchaseOrderDetail ppod
JOIN Production.Product pp
ON ppod.ProductID = pp.ProductID
JOIN Production.ProductSubcategory AS PSC
ON PSC.ProductSubcategoryID
   = pp.ProductSubcategoryID
JOIN Production.ProductCategory AS PC
ON PC.ProductCategoryID = PSC.ProductCategoryID
ORDER BY 1
```

Figure 4 SQL statement for populating the identified mining structure

Clustering analysis is suitable for customer segmentation. Given that the relationship between Customer and Product Purchased are one to many, the target mining structure for clustering should contain a nested table column on products for each customer, as shown in Table 3.

PurchaseOrderID	TotalDue	ShipMethodID	Status	VendorName	CreditRating	VendorStatus	ActiveFlag	Product	Category	ubCatego	Color	ListPrice	Size	hitMeasur	Weight	UnitMeasu	Class	Style
1	222.1492	3	4	Litware, Inc.	1	1	1											
2	300.6721	5	1	Advanced Bicycles	1	1	1											
3	9776.267	2	4	Allenson Cycles	2	1	1											
4	189.0395	5	3	American Bicycles and Wheels	1	1	1											
5	22539.02	4	4	American Bikes	1	1	1											
6	16164.02	3	4	Anderson's Custom Bikes	1	1	1											
7	64847.53	3	4	Proseware, Inc.	4	0	0											
8	766.1827	5	4	Aurora Bike Center	1	1	1											
9	767.0528	5	4	Australia Bike Retailer	1	1	1											
10	1984.619	5	4	Beaumont Bikes	1	0	1											
11	553.8221	4	4	Bergeron Off-Roads	1	1	1											
12	38281.87	5	4	Bicycle Specialists	1	1	1	Touring Pedal	Components	Pedals	Silver/Black	80.99	NULL	NULL	NULL	NULL	NULL	NULL
13	2032.654	4	4	Bike Satellite Inc.	1	0	1											
14	161.6461	5	3	Bloomington Multisport	1	1	1											
15	113.3332	5	4	Burnett Road Warriors	1	1	1											
16	166.6235	5	4	Business Equipment Center	2	1	1											
17	15104.71	5	4	Capital Road Cycles	1	1	1											
18	18114.52	5	4	Carlson Specialties	2	1	1											
19	87520.56	2	4	Chicago City Saddles	1	1	1	ML Mountain Seat/Saddle	Components	Saddles	NULL	39.14	NULL	NULL	NULL	NULL	M	NULL
								HL Mountain Seat/Saddle	Components	Saddles	NULL	52.64	NULL	NULL	NULL	NULL	H	NULL
								LL Road Seat/Saddle	Components	Saddles	NULL	27.12	NULL	NULL	NULL	NULL	L	NULL
								ML Road Seat/Saddle	Components	Saddles	NULL	39.14	NULL	NULL	NULL	NULL	M	NULL
20	609.8274	1	4	Chicago Rent-All	2	1	1											
21	7721.464	1	4	Circuit Cycles	1	0	0											
22	31019.86	2	4	Comfort Road Bicycles	1	1	1											
23	41230.06	1	4	Compete Enterprises, Inc	1	1	1	HL Road Pedal	Components	Pedals	Silver/Black	80.99	NULL	NULL	149	G	H	NULL

Table 2. Target data view for classification

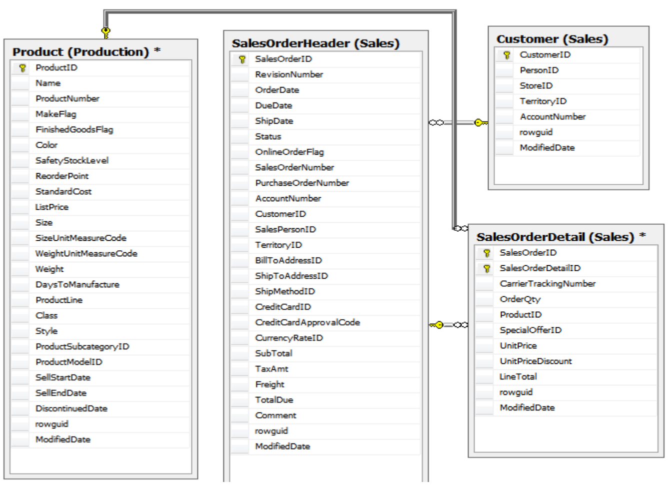


Figure 5. Schema for tables needed for clustering

CustomerID	TotalAmount	CustomerType	ProductName
11000	9115.1341	B	Sport-100 Helmet, Red
			Mountain-100 Silver, 38
			Mountain-200 Silver, 38
			Fender Set - Mountain
			Short-Sleeve Classic Jersey, S
			Touring Tire Tube
			Touring Tire
			Touring-1000 Blue, 46
11001	7054.1875	B	Short-Sleeve Classic Jersey, XL
			Fender Set - Mountain
			Mountain-200 Silver, 38
			Mountain-100 Black, 44
			Sport-100 Helmet, Black
			AWC Logo Cap
			Water Bottle - 30 oz.
			Water Bottle - 30 oz.
			Mountain Bottle Cage
			Road Bottle Cage
11002	8966.0143	B	Mountain-200 Black, 38
			Sport-100 Helmet, Blue
			Mountain-100 Silver, 44
			Touring-1000 Yellow, 46

Table 3 Target data view for clustering

```
SELECT sc.CustomerID, sum(soh.TotalDue) AS TotalAmount,
        CASE WHEN sc.StoreID != null THEN 'S'
        WHEN sc.PersonID != null THEN 'I'
        ELSE 'B' END AS CustomerType
FROM Sales.Customer sc
JOIN Sales.SalesOrderHeader soh
ON sc.CustomerID = soh.CustomerID
GROUP BY sc.CustomerID
GROUP BY 1

SELECT  sc.CustomerID, pp.Name AS ProductName
FROM Sales.Customer sc
JOIN Sales.SalesOrderHeader so
ON sc.CustomerID = so.CustomerID
JOIN Sales.SalesOrderDetail sod
ON so.SalesOrderID = sod.SalesOrderID
JOIN Production.Product pp
ON sod.ProductID = pp.ProductID
ORDER BY sc.CustomerID
```

Figure 6. SQL statement for populating the target data view

The population of the identified mining structure needs to include two SQL statements as shown in Figure 6. The second SQL statement in Figure 6 is used to populate the nested table column of the identified mining structure shown in Table 3.

6 The Proposed Use Case on Time Series Forecasting

The business need for this use case is to generate a mining model that is able to predict the sales amount for a specified product category in future months. The relevant information for this analysis includes product and sales order. We identify the following source tables as shown in Figure 7. We further specify the suitable target data view for time series forecasting as shown in Table 4.

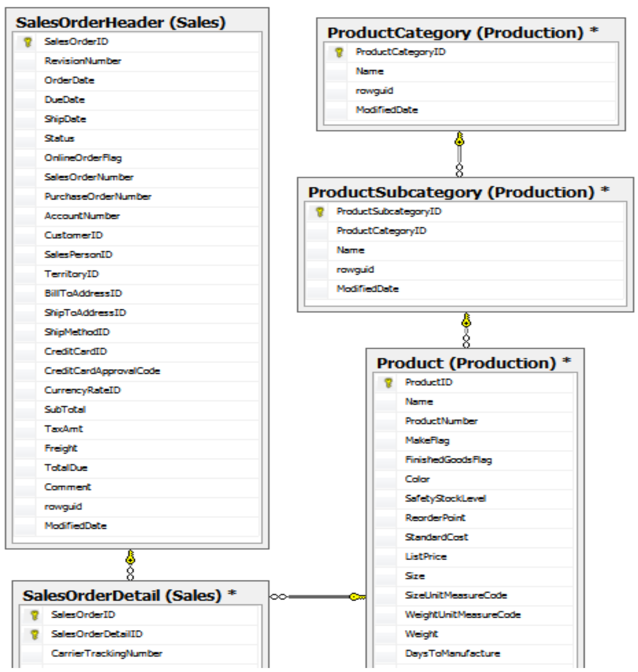


Figure 7. Schema for tables needed for time series prediction

ProductCategory	OrderYear	OrderMonth	TotalDue
Components	2001	7	3812631.065
Components	2001	8	6846445.728
Components	2001	9	5114408.463
Components	2001	10	8755902.449
Components	2001	11	19929188.27
Components	2001	12	14443257.87
Components	2002	1	6658367.452
Components	2002	2	17146129.03
Components	2002	3	11967565.47
Components	2002	4	8612439.03
…	…	…	

Table 4 Target data view for time series prediction

The query that is needed to populate the target data view from the identified source tables is presented in Figure 8.

```
SELECT pc.Name,
        YEAR(soh.OrderDate) as OrderYear,
         month(soh.orderDate) as OrderMonth,
        soh.TotalDue
FROM Sales.SalesOrderDetail sod
JOIN Sales.SalesOrderHeader soh
ON sod.SalesOrderID = soh.SalesOrderID
JOIN Production.Product pp
ON pp.ProductID = sod.ProductID
JOIN Production.ProductSubCategory psc
ON pp.ProductSubcategoryID = psc.ProductSubCategoryID
JOIN Production.ProductCategory pc
ON psc.ProductCategoryID = pc.ProductCategoryID
WHERE ProductCategoryID = 2
```

Figure 8. SQL statement for populating the target data view

7 Conclusions

Based on the enterprise level sample database AdventureWorks, we proposed four comprehensive use cases that enable students to conduct data mining in a comprehensive business data environment with the goal of meeting specified business requirements. The proposed use cases cover the following most frequently used data mining techniques: association mining, classification, clustering, and time series prediction. Each use case is described in details from the following perspectives: specifying business requirement, locating source tables, identifying suitable mining techniques and the target data view or mining structure, populating the target data view or mining structure. The proposed use cases can be easily tailored to fit in hands-on assignments or comprehensive course project in a data mining course. The future work on this project includes studying the effects of these use cases on learning outcomes of our IS data mining course.

References

[1] Gorgone, John; Davis, Gordon B.; Valacich, Joseph S.; Topi, Heikki; Feinstein, David L.; and Longenecker, Herbert E. "IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems," *Communications of the Association for Information Systems*: Vol. 11, Article 1. Available at: http://aisel.aisnet.org/cais/vol11/iss1/1, 2003

[2] Han, Jiawei; Kamber, Micheline. *Data Mining: Concepts and Techniques*, Elsevier, ISBN 1558609016, 2006

[3] MacLennan, Jamie; Tang, ZhaoHui; Crivat, Bogdan. *Data Ming with Microsoft SQL Server2008*, Wiley Computer Publishing, ISBN: 9780470277744, 2009

[4] Suh, Sang. *Practical Applications of Data Mining*, Jones & Bartlett Learning, ISBN 9780763785871, 2011

[5] Linoff, Gordond; Berry, Michael. *Data Mining Techniqeus: for Marketing, Sales, and Customer Relationship Management*, Wiley Computer Publishing, ISBN: 9780470650936, 2011

An empowering *Introduction to Computer Science* course for general education that teaches the fundamentals

L. Liao
Department of Mathematics and Computer Science, University of San Diego, San Diego, CA, USA

Abstract - *This paper describes a general education Introduction to Computer Science course based on a computer science course for non-majors that I have taught almost non-stop since 1983 with great success at the University of San Diego. The course teaches the fundamentals of computer science. It is organized around three topics, in this order: effective use of applications; PC, networking and client-server technology; and programming. It empowers students with new tools and new ways of thinking to solve problems. The paper concludes with a comparison with similarly named courses in two Taiwan universities that have little in common.*

Keywords: general education; application; networking; client-server; programming; Python

1 Introduction

Common things computer users do when something undesirable happens are

- Reboot the computer. For example, reboot a computer when it loses connection to the campus network.
- Keep on trying, wishing the problem goes away. For example, keep on requesting a web page that will not load.

The *keep-on-trying* method is as good as burning incense to please the computer god. The *reboot* method is tragic-comical if the problem is caused by a server machine taken off-line or a server program quitting unexpectedly. The prevalence of these mindless approaches of dealing with computing problems suggests a general lack of computer literacy and hence the importance of a computer science general education requirement.

This paper describes an *Introduction to Computer Science* general education course based on a computer science course for non-majors that I have taught almost non-stop since 1983 at the University of San Diego [2]. It teaches the fundamentals of computer science and at the same time empowers the students. The course is organized around three topics, in this order:

- Effective use of applications
- PC, networking and client-server technology, and
- Programming

These components are obviously not mutually exclusive. On the contrary, they are intimately related. First, without programming, there would not be applications and client-server technology. Next, understanding of programming and client-server technology definitely helps with use of applications because one would reason and follow logic based on understanding, instead of relying on brute-force memorization.

The course empowers students with new and powerful tools such as browser extensions, and new and powerful thinking such as programming. It demonstrates how such tools and thinking help them solve problems they could not otherwise. We do not ask students to memorize things they consider irrelevant.

At a bus top in Taichung, Taiwan, I asked a student how long she had been waiting for the bus under the hot September sun. She replied, “More then ten minutes.” I showed her my iPhone, which displayed a web page showing that the bus was coming in just now, and said, “Get an iPhone or an Android phone and you won’t need to wait for any bus in the dark”. Who would want to wait for buses in the dark everyday if one can know when the buses come? Likewise, who would continue to check for new Gmail messages, when one could be notified by a Gmail notifier extension? When you teach students powerful things such as these, the students will embrace them with enthusiasm.

In III, comparisons are made with courses that are titled identically or similarly such as Introduction to Computing, but otherwise have little in .

2 The three components of the course

I start the course with applications students believe they already know very well such as browsers, email clients, Microsoft Word and Excel. Many students use applications knowing the bare minimum, such as *type-and-save* in the case of Microsoft Word and Excel. Showing them how such familiar applications can help them solve problems in ways they did not know before will change their habits and attitude with computers. We should teach not just new *features and tools*, but also new ways to *think,* such as programming. Other courses teach students to use Microsoft applications; we teach students to consider alternatives such as Google apps and Evernotes [3], in addition to and instead of them.

2.1 Use applications effectively

I cover the following in the first part of the course. I present them here hoping they will be helpful to others.

2.1.1 Email

People complain about junk mails or excessive volume of emails; few take actions. Most would manually delete messages, but would not set up filters to *automate* deletion, archiving or forwarding, as appropriate. They would not unsubscribe an unwanted mailing list; elect instead to delete such messages one by one with anger. Some unsubscribe important mailing lists because they do not know other alternatives such as setting up their accounts to receive a digest daily, or no messages ever.

It is useful to design exercises to make sure that students know the basics of mail folders, labels, mailing lists, and the pro and cons and working knowledge (including server setup) of POP, IMAP, and web mail.

Many use emails to send files or to gather data, when other methods should be used. See 2.1.4 for details.

2.1.2 Browser

Recently, I asked a group of twenty students to list the most important features of a browser. Add-ons/extensions were never mentioned. When I later demonstrated the power of browser extensions, I could see the storms in all these students' brains. Here are some add-ons that are part of my daily life:

- Answer [5]: one-Click on a word in a web page shows information about that word, in context.
- Firebug [6]: allows one to edit, debug, and monitor CSS, HTML, and JavaScript live in any web page.
- Google Notebook Extension [7]: select and click to copy a portion of a web page, as well as the URL of the page, into a Google notebook.
- WorldIP [8]: shows up-to-date information about real physical location (country, flag, network, Provider/Datacenter and IP address, reverse DNS) of a web server you are currently visiting, as well as display your own IP address and service provider.

Browser extensions provide useful and practical examples of extensibility, an important concept of computer science.

2.1.3 Publishing to the Web

Nowadays, it takes little effort or expertise to be a web producer, using nothing more than a browser. We can turn a student into a powerful web publisher in hours, by showing her how

- To set up a Blogspot blog. Then teach her to add graphics, and do simple formatting such as quotation and bullet list. Teach her to use the HTML editor to learn HTML quickly, to embed objects such as videos and Google maps, and to switch to HTML mode when graphic mode becomes unhelpful.
- To create Google sites to publish web sites with complex structure. At this point, students can publish files of any type by simply uploading them to a Google site and to write a book with automatically generated table of content.
- To make use of Google spreadsheets to create interactive web pages that take user inputs and display them selectively.
- To use Google documents, spreadsheets, calendars for collaboration.

Here in 2.1.3, students use browser to produce web sites, not to browse them. Invite them to pause to ask why the program that does so much more than just allowing people to browse the web is called a *browser*. A better name is *user agent*.

Note that this section can be covered in the second part of the course after students have more exposure to client-server technology.

2.1.4 Use Correct Software

Examples topics:

- Do not use Microsoft Word document to store a database table; use a DBMS such as MySQL or Postgresql instead.
- Do not use spreadsheets to store a database; use a DBMS instead.
- Do not use email to collect and publish data, such as petitions, use web applications instead.
- Instead of emailing pictures or other files as attachments, consider publishing them to web sites, and emailing just links to them.

2.1.5 Look for opportunities to automate

Lead students to seek automation when the job involves repetition or contains common elements that could be factored. The following examples may be helpful.

- Routine email processing as described in 2.1.1
- Sketch the graph of $3x^2 - 5x + 7$ on the interval [0, 1] using 1001 grid points. A student may not think about automation when he enters the first few grid points 0, 0.001, 0.002, ..., by hand. It is a sure bet that most will soon. In the unlikely event that a student actually enters all 1001 grid points by hand, the experience could be inspirational.
- Sort a list of names of the form '*first_name last_name*' in the alphabetical order of *last_name*.
- Write job application letters to 200 companies. Two letters differ only in the company information such as company names, addresses, etc. Do not tell students that this can be easily accomplished with mail merge. Give them the pleasure of finding this out themselves.
- You make tea daily and like to soak your tea bag in hot water for exactly three minutes. You should be looking for a timer program that can countdown to 0. If you cannot find one, you must sharpen your search and research skill in using search engines. Consider writing a timer program yourself. Smartphone users should find such programs a part of their lives.
- Wake yourself up everyday at 6 AM using your computer. Smartphone users should find such programs a part of their lives.
- Schedule your computer to run a program at a set time everyday.

This marks the end of the first part of the course that turns students into power users relative to what they were when they started. At this point, students use familiar programs productively with a fresh pair of eyes. They use browsers with diverse extensions that they did not know existed. They use Microsoft Excel with programmers' insight: writing formulas, propagating them, splitting a column to separate one entity into many, with follow-up sorting. They learn to adjust the behavior of programs by configuring them. Our constant use of Google search in class also helps students' search and research ability.

Let us move on to the second component.

2.2 PC, Networking and Client-Server technology

I start an introduction of networking with the machine in front of the student: her PC.

2.2.1 PC

My class finds the following lesson on PC hardware fun and instructive. In a class of twenty, I would bring two PCs and a good number of screwdrivers, and

- Ask students who never before opened a computer case to open up a PC
- Ask students to identify major components of a PC: motherboard, CPU, memory, daughter boards, hard drive, and network interface card, etc.

Now is a good time to point out that

- Having an inadequate amount of memory and hard drive space is like locking a giant in a closet. It is often the reason PCs with powerful processors perform sub-optimally [4].
- Related to the above is the very common habit of closing a program, for example, a browser, when it is not in use. But the program is opened soon again. People should stop engaging in this perpetual cycle of opening and closing programs, and instead, maximize their PC memory so as to run many programs concurrently.
- A mouse is the slowest device of a computer, so that one should get into the habit of using keyboard shortcuts, such as control-C for copy, control-V for paste and control-K for line deletion, instead of the slower mouse operations.

2.2.2 Networking and client-server technology

Assuming students' computers, be they lab computers or their own notebook computers, are connected to the campus network, we can devise simple exercises to illustrate the power of network connectivity. One useful exercise is to ask students to configure their PCs to print to two nearby network printers, preferably in different buildings. This leads to the natural topic of router configuration to allow all home PCs to print to a network printer at home [4].

I assign chapters from Doug Comer's *The Internet book: Everything You Need to Know About Computer Networking and How the Internet Works* [1] as reading assignments. My students find the book highly readable and valuable. In addition, I can suggest the following topics:

- Teach networking and router basics. An Internet service can stop working for many different reasons. The problem could be user's phone line, DSL or cable modem, ISP, server, firewall setting, or software failure. Frequently a user is frustrated when unable to access a service and looks in all the wrong places due to lack of understanding. A dramatic example was a student who called her DSL service provider using her cell phone when her browser was not able to access a web page because her phone line was dead. Teaching students networking and router basics can make them better troubleshooters.

- Discuss different storage models of applications. For example, Microsoft Word (without SharePoint) saves files to user PC's hard drive. This contrasts with Google's server-side storage model and Evernote's client-server storage model. This means, when you edit a Google document, the document is stored on Google's server; and when you edit an Evernote, it is saved to both your hard drive and Evernote's server. These three models provide useful and practical examples of client-server technology. Understanding this can help students avoid the common pitfall of being unable to access a Word file while in school and understand why Google docs are better suited for collaboration.
- Teach students to use terminal program to issue shell commands; to talk to a server program using "*telnet server_name port_number*"; to use secure shell ssh interactively. This makes such tools as ping, ifconfig, nslookup, whois, wget, traceroute, etc available from the command line.
- Teach students to publish to the web as described in 2.1.3

The first two parts of the course lead naturally to the question: *What does it take to produce such applications?* The simple answer is *programming*, which is the last part of the course. I save programming for the last not just because I want the powerful applications used before to motivate students to want to program. Programming is fundamental, but is also most difficult to motivate in a general education course. We must convince students that even if they do not become programmers, knowing programming is useful and makes them better persons.

2.2.3 Programming

Instructors may choose not to teach programming in a general education computer science course, because they believe students

- With weak background in math and science may not have the mental capacity to learn programming.
- Could not write useful programs based on what they can reasonably learn in such a course, and thus the effort is futile, and the time can be better spent on other topics.

I offer a different perspective:

- Choosing a language such as Python can make teaching and learning programming possible, and
- After learning conditional and looping, students are in a good position to understand the following:
  - An ATM machine runs a loop with enclosing conditionals.
  - A mail-merged document is created with a loop with variables substituted by values from a data source, such as a spreadsheet.
  - An Excel formula copy-down is accomplished with a loop.
- Programming sharpens students' sensibility about automation, whether from configuring programs or by writing code.

I strongly recommend Python because it is

- Easy to learn
- Hard to forget, and
- Elegant

On a personal note, programming in Python makes me happy.

Programs we write in my class include an alarm clock, an arithmetic tutor and a crude blackjack program. In [2], I say this about Python:

> We can do so mach in such a short time only because the Python programming language does not get in the way of the programmers, thus allows them to concentrate on programming.

3 A comparison with other courses with identical or similar names

Two types of courses are popular. One has the appearance of training students to use certain applications; the other is a survey of selected computing topics. A combination of training and survey is also common. An example training-survey course is found at Taiwan's Tamkang University (淡 江 大 學) at http://www.scribd.com/doc/23112709/資訊概論主課程與實習課程綱要.

In this particular course, the applications used include

- Microsoft Internet Explorer, a browser
- Microsoft Word, a word processor
- Microsoft Excel, a spreadsheet program
- Microsoft PowerPoint, a presentation software
- Adobe Dreamweaver, a popular program for web page construction
- Adobe PDF Reader

I do not see much computer *science* education in this list. The supporters of training courses say they give students useful skills to make them employee-able. Opponents argue that such courses may be appropriate in professional schools such as a plumbing school, but not in a liberal arts university whose mission is to teach the ability to learn, not specific technical skill.

Another popular approach is to survey selected topics in computer science. An example course is http://www.csie.ntu.edu.tw/~kmchao/bcc07spr/index.htm.

A problem with this approach is that while it might awe students with the breadth of computer science, it could not have depth and it could not give students real understanding or working knowledge. For this particular course of Taiwan University (臺灣大學), one of the best universities in the world, I have many reservations:

- When the students do not know what programming is, what do we expect them to gain from a survey of programming languages, programming paradigms, and parameter passing mechanisms?
- Survey of Operating Systems and Networking feels like an effort to condense and simplify textbooks on such topics into a few pages to be covered in a few classes. Given the little computing sophistication of the students, what is the expected learning outcome?

Programming, data structures and algorithms are taught using the system programming language C, a programming language unsuitable for general education, even for the best of students.

4 Conclusion

This paper describes a general education *Introduction to computer science* course that empowers the students and at the same time teaches the fundamentals of computer science. I have taught a similar course for non-majors at the University of San Diego for 17 years. My course has inspired many students to pursue a computer science major. I believe it has empowered *all* students who have taken the course.

5 References

[1] Doug Comer. "The Internet book: Everything You Need to Know About Computer Networking and How the Internet Works". Prentice Hall, 2006.

[2] Luby Liao and Jack Pope. "Computer Literacy for Everybody"; Journal of Computing Sciences in Colleges, 23-6, 231-238, 2008.

[3] Luby Liao, "Microsoft and I"; http://comp-lite.blogspot.com/2010/08/microsoft-and-i.html.

[4] Luby Liao, "What price computer illiteracy"; http://comp-lite.blogspot.com/2008/08/what-price-computer-illiteracy.html.

[5] Answer browser extension at https://addons.mozilla.org/en-US/firefox/addon/735/.

[6] Firebug browser extension at https://addons.mozilla.org/en-US/firefox/addon/1843/.

[7] Google Notebook browser extgension at https://addons.mozilla.org/en-US/firefox/addon/50380/.

[8] WorldIP browser extension at https://addons.mozilla.org/en-US/firefox/addon/8661/.

Building the Biological and Statistical background for Bioinformatics in Engineers at UPRM

M. Cabrera-Ríos[1], J. Seguel [2], and P. Ortiz Bermúdez[3]
[1]Industrial Engineering Department, University of Puerto Rico - Mayagüez, Mayagüez, PR, USA
[2]Electrical &Computer Engineering Department, University of Puerto Rico - Mayagüez, Mayagüez, PR, USA
[3] Department of Chemical Engineering, University of Puerto Rico - Mayagüez, Mayagüez, PR, USA

Abstract – *Bioinformatics is a field that requires its professionals to have a level of expertise in multiple disciplines in order to contribute in practice and research. In recognition of this fact, the College of Engineering at UPRM has set out to train a new generation of Engineers capable of tackling bioinformatics problems with mathematical and statistical proficiency, computational efficiency, and biological know-how. This work describes two courses, one in Quantitative Biology and the other in Statistics for Bioinformatics that will be part of a curricular sequence proposed to this end.*

Keywords: Bioinformatics, Statistics, Quantitative Biology, Multidisciplinary Approaches

1 Introduction

Interdisciplinarity is posed to only gain importance as a driver for development and discovery. Fields like Bioinformatics will capitalize on specialized expertise from different disciplines to further knowledge about important health problems. In order to successfully contribute in an interdisciplinary field, there must be a minimum degree of formal and functional understanding of all related disciplines.

In the College of Engineering at UPRM, a Minor in Bioinformatics has been proposed within its curricula [1] precisely in recognition of the value of blending the traditional solution-driven pragmatism in Engineering with the biological insight necessary to tackle the large challenges of the life sciences.

Among the disciplines embedded in the above-mentioned minor, and thus in the area of Bioinformatics, the effective integration of biological, statistical, and computational fundamentals takes a paramount role. This report presents the creation of two new courses, which attempt a functional and focused approach in the incorporation of biological and statistical frameworks into the profile of a bioinformatics specialist.

2 Integration of Statistics, Biology, and Bioinformatics

Although it is not uncommon for Engineers to have training in statistics, the fact that Bioinformatics demands the understanding of biological principles to solve particular kinds of problems requires that a focused point of view in statistics be embraced in a new course; one that matches the relevant biological problems with the adequate statistical technique.

What sets bioinformatics apart from most computational science fields is the fact the laws of biological information processing and phenotype expressions, as well as those that govern cell signaling, are mostly of a statistical nature. "Biology has rapidly become a large source of new algorithmic and statistical problems, and has arguably been the target for more algorithms than any of the other fundamental sciences" (N. Jones, P. Pevzner). For instance, methods for probabilistic and statistical data analysis are the essence of microarray bioinformatics.

Computational probabilistic and statistical methods play a central role in phylogenetic inference and in sequence alignments, as well. BLAST, a well – known computer package for sequence analysis, is based on an elaborate probabilistic model. Statistics plays also a crucial role in the design and validation of bioinformatics experiments. Indeed, a bioinformatics expert cannot rest on deterministic mathematical models for the design and validation of his/her information processing methods, as a computational physicist, or a computational chemist may do. As a consequence, the design of effective bioinformatics methods requires a well – designed educational path in applied probability and statistics, in addition to training in the engineering of computing systems. The relationship between computing and scientific methods in life sciences is, in most instances, so tight that often the computing method is almost indistinguishable from the scientific method itself. Consequently, in most practical situations, bioinformatics specialists must work in close collaboration with life scientists in the planning of experiments and in the collection and organization of data.

This is often the only way to maximize the production of reliable and useful information. Bioinformatics specialists are crucial to the elucidation of bio – data processing requirements, development and implementation of new bio – information processing methods, and the analysis and interpretation of computer generated biological results.

These tasks cannot be accomplished without a breadth of knowledge of molecular genetics and structural biology. Although not an expert in life sciences, a bioinformatics specialist must understand and speak fluently the language of geneticists and cell biologists. As stated before (Doom, T., et al., 2003), an attempt to accomplish the successful integration of Biology into a Bioinformatics undergraduate curriculum, should focus on teaching "genetics, molecular and cellular biology, chemical and physical aspects of the flow of genetic information from DNA to proteins, gene expression, replication, recombination and repair, and the experimental tools of molecular biology" [2]. In addition, these concepts should be put in the context of the corresponding quantitative tools available to date to foster the development of knowledge. In this manner, effective correlations between the descriptive character of biosciences and the quantitative mindset of a bioinformatics specialist will be established.

3 A course in Quantitative Biology for Engineers and Computer Scientists

The Department of Chemical Engineering (DChE) at the UPRM has fostered the design of a course to provide fundamental and functional biological concepts to engineers and computer scientists. This course is directed towards freshmen/sophomore level undergraduates pursuing a bachelor in engineering. The course is entitled "Quantitative Biology for Engineers" and as its title suggests, will introduce basic biology concepts emphasizing on specific quantitative tools used to date to address problems and apply knowledge. This course is part of the Bioinformatics Minor currently being developed at the UPRM.

The DChE has allocated significant resources in developing bio-based research and education, recruiting professors who have been trained formally in a biology-related field. This is a response towards the current trends in the chemical engineering discipline, which has increasingly become a cornerstone in the solution of biomedical/bioprocess problems. The need to train chemical engineers at the undergraduate level with basic biological notions became imperative and as such the DChE has responded proactively. The course presented here is a result of these efforts.

The necessity of such a course transcends chemical engineering and coincided with the development of the Bioinformatics minor at the UPRM. Undoubtedly, this course will have tremendous impact in our institution and beyond, given the fact that our engineering school trains most of the professional engineers in our country.

The Quantitative Biology for Engineers course will cover fundamental concepts and language of biology, emphasizing on basic molecular notions from an engineering perspective. Among the topics included in the course are: the functions of life, components of a biological cell, the genetic code and the flux of genetic information, protein structure and function, concepts of systems biology and ecosystems. A relatively novel approach for the implementation of this course is the introduction of quantitative tools such as MATLAB in biological scenarios providing examples of how such quantitative tools can be applied to a bio-based question related to medicine and engineering. In the next section specific examples of this proposal will be discussed. An outline of the course is presented in Table 1 shown below.

Table 1. "Quantitative Biology for Engineers" course outline

Outline	Contact Hours
Biological cell structure and physiology	4
Genetic code for information transfer	6
Genomics, proteomics, transcriptomics	6
Bioenergetics	1
Metabolic Networks and Enzymes	6
Cell-cell interactions	6
Biological defense mechanisms: molecular immunology	3
Bioethics	1
Applications	6
Project presentations	6
Total hours: (equivalent to contact period)	45

The assessment tools that will be utilized to evaluate the course are the following: 3 partial exams (50%), quizzes and assignments (12.5%), 1 project (12.5%) and a final exam (25%). According the assessment tools planned for this course the ABET outcomes that will be measured are shown in Table 2.

Table 2. ABET outcome assessment criteria measure in the course.

Assessment Criteria	Assessment Tool
a. An ability to apply knowledge of mathematics, science, and engineering	Exams
b. An an ability to design and conduct experiments as well as to analyze and interpret data	Assignments using genetics and biochemistry simulators and MATLAB to solve problems and analyze a data set
d. An ability to function on multi-disciplinary teams	Course project
e. An ability to identify, formulate and solve an engineering problem	Course project
f. An understanding of professional and ethical responsibility	Case study assignment/quizz on

	bioethics
g. An ability to communicate effectively	Course project presentations
h. The broad education necessary to understand the impact of engineering solutions in a global and societal context.	Course project
j. The knowledge of contemporary issues	Course project

4 Teaching Biology to Engineers and Bioinformatics Specialists: Integration of Quantitative Tools

One of the effects of technological advances in molecular and systems biology is the generation of massive amounts of data. The challenge then became the management and analysis of such data sets. This not only gave rise to the birth of a new field, bioinformatics, but initiated the transition of the way biology is perceived and taught in the classroom. Biology has evolved from a mostly descriptive and observation-based discipline to one, which utilizes computation, statistics and mathematics to explain nature. Gradually, this new profile of a biologist is being reflected in undergraduate education.

The Quantitative Biology for Engineers course has been designed to incorporate quantitative tools into basic biology formation targeted to engineers and bioinformatics specialists. For instance, basic concepts of biochemistry such as protein function and structure can be presented using visualization tools such as Pymol, StarBiochem and Fold It (open source developed at MIT). Particularly, with Fold It students will be able to apply concepts of protein structure to solve puzzles based on the principles that govern protein folding. While introducing enzymes and protein binding concepts, MATLAB can be used to calculate kinetics and equilibrium constants. In genetics, simulators such as StarGenetics can be used to simulate genetic crosses and design experiments. This is particularly useful to reinforce basic concepts such as dominance, recessivity, linkage and complementation. In systems biology and molecular biology, MATLAB can be used to establish correlations among large quantities of data to determine interactions between genes or metabolic networks, and to analyze gene sequences [3]. Similarly, biological data such as mutant analysis data can be simulated in R. All these tools will be introduced in the course to give a flavor of tools, which will be studied in depth in more advanced courses within the Bioinformatics minor at the UPRM.

5 A course of Statistical Methods in Bioinformatics

The Department Industrial Engineering at UPRM will organize this course within the minor in Bioinformatics. The course, Statistical Methods in Bioinformatics, capitalizes on the Department´s incursion into biological data analysis through the Bio IE lab and its well-established curriculum in probability and statistics.

This course, designed for 3 credit hours, involves the study and application of statistical techniques to several of the most important bioinformatics analysis including: sequence analysis, BLAST, microarray analysis, gene finding, phylogenetic trees and analysis of evolutionary processes.

In putting together this course, an exercise of matching the above-mentioned analyses to statistical techniques was carried out. This exercise resulted in identifying inferential statistics, hypothesis testing, probabilistic estimation, clustering analysis, and markovian processes as the most important techniques to be covered.

As a prerequisite, it is envisioned that the students have already taken a course in Probability and Statistics equivalent to that taught by the Industrial Engineering department (ININ 4010). In this course, the basics of probability are discussed, to then cover the topics of probability distributions (discrete and continuous), to finally explore the concepts associated to inferential statistics: parameter estimation, confidence intervals and hypothesis testing.

An important trait of the curricular sequence is the idea of keeping a hands-on approach to teaching. To this end, the course starts with a review of probability and statistics as well as an introduction to stochastic processes to then approach each of the topics, organized by analysis task, as a whole project. That is, each topic will start with an objective for the analysis to then gather the right data, selecting the adequate statistical tools, obtaining statistical outputs to finally interpret these outputs in the biological context. Table 3 shows the topic sequence.

Table 3. Topic sequence

Outline	**Contact Hours**
Review of Probability and Statistics	5
Introduction to Stochastic Processes	5
Sequence Analysis	6
BLAST	6
Microarray Analysis	6
Evolutionary Processes	6
Phylogenetic Trees	6
Total hours: (equivalent to contact period)	40

The course will be evaluated with three partial exams accounting for 60% of the final grade, a final exam (25%), a final project (10%) and attendance (5%). An example of how each bioinformatics task will be covered is shown next.

6 Teaching a particular topic: Microarray Data Analysis

Microarrays are high-throughput experiments capable of measuring the relative expression of tens of thousands of genes in a simultaneous manner. Microarray analyses are usually carried out with the common purpose of detecting a particular behavior on gene expression. One of the most common objectives is that of selecting genes that show a significantly different expression behavior when measured in tissues in two distinct states, such as, healthy tissues and tissues with cancer. This objective, also called gene selection or filtering, can be used to guide our analyses.

One first step in this topic is to show the students to obtain data from public repositories such as GEO [4]. Once with a potential data set, a series of descriptive statistics can be applied to explore the data and detect potential problems on dimensionality or due to outliers.

Data from microarray experiments have been shown to have a large and nonconstant variance across samples as well as not follow normality. Furthermore, because these experiments are still expensive, most datasets do not have a number of replicates large enough to justify a normal approximation. The students will be guided with statistical tools for them to experience and detect these behaviors with the selected database.

With the characteristics of the data set, a series of potential statistical tools will be presented to the students. The instructor will teach the main assumptions, capabilities, mathematical structure, and execution of each tool. The student will be left with the task of running a comparative analysis across these tools to determine the one with the best performance for gene selection. Where applicable, a statistically designed experiment will be encouraged.

The genes that are finally deemed important by the statistical analysis will be then used to search the roles of these genes using online biological tools such as Gene Ontology [5].

7 Conclusions

Although born out of a need for storing and organizing huge amounts of biological data, Bioinformatics has evolved rapidly over the years into a much more comprehensive discipline. The emerging disciplinary character of Bioinformatics shifts its original database centric character to a much more integrative, multidisciplinary character in which statistics and biological systems modeling play equally important roles. Such emerging character and the need for professionals endowed with a well-rounded training opens up opportunities for a much wider range of disciplines to join the fascinating search for the principles and laws of life sciences and its multiple applications. Their training in the modeling of multiple phenomena and in the design and test of systems solutions makes Engineering students especially suited to join Bioinformatics research and application efforts. This manuscript describes two courses: one on statistical techniques for bioinformatics and another on quantitative biology, proposed for a curricular sequence in Bioinformatics within the College of Engineering at UPRM. The course Statistical Methods in Bioinformatics will be application-driven with an emphasis on adequate statistical tool selection for a particular analysis objective in mind. It will be important for the students enrolled in this course to have an introductory course in Probability and Statistics. It is expected that this course provide the students with the capability to understand statistical processes and to carry out comparative experiments to determine the best-suited solutions –from a statistical point of view-. The students are also expected to validate these solutions with biological information to complete the interdisciplinary cycle.

The course Quantitative Biology for Engineers focuses on the formation of a basic biological framework for engineers and bioinfomatics specialists. It will focus on molecular aspects of biology overlapping fields such as genetics, physiology, biochemistry, and physics. A particular attention will be placed in showcasing quantitative tools currently being applied in the biosciences. Students will have hands-on experience with these tools and assignments will be directed towards their utilization.

8 Acknowledgement

This work was made possible thanks to the NIH-MARC grant “Assisting Bioinformatics Efforts at Minority Institutions” PAR-03-026 and BioSEI UPRM grant 330103080301.

9 References

[1] J. Seguel and M. Cabrera-Rios, A Minor in Bioinformatics within the Engineering Curricula, 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'09: July 13-16, 2009, USA) 68-74

[2] T. Doom, M. Raymer, D. Krane, and O. Garcia. 2003. “Crossing the Interdisciplinary Barrier: A Baccalaureate Computer Science Option in Bioinformatics” IEEE Transactions on Education, 46 (3):387-93

[3] J. J. Cai, D. K. Smith, X. Xia and K. Yuen. 2005. “MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution”, *BMC Bioinformatics*, 6 (64)

[4] http://www.ncbi.nlm.nih.gov/geo/.

[5] http://www.geneontology.org/

Beyond Programming and Software Development: Additional Teaching/Learning Goals throughout the Curriculum

Donald R. Schwartz
Department of Computer Science
Millsaps College
Jackson, MS USA

Abstract - *As with most colleges, we always believed that our Computer Science program was strong and that our graduates performed well in the "real world". Feedback from alumni informed us that some were at a loss when performing "basic" activities like completing self-assessment reports for annual evaluations or reports in which they assessed other workers, either peers or those they supervised. A few students who went to Graduate School informed us that they were taken aback at the amount of material the professors expected them to cover on their own. Informal feedback such as this took us a bit by surprise, but over the past decade, we have made great strides in better preparing our students for "life after college". This paper describes a few of the successful curricular enhancements we have made to address these deficiencies, focusing on both upper-level and lower-level courses.*

Keywords: *Curriculum Development, Preparing graduates for academia, Preparing graduates for industry, Evaluation strategies, Transition to graduate studies*

1 Introduction and Philosophy

I believe that students should be considered both clients and active participants in their education. Students come to us for a first-class education – it is our responsibility to provide the means necessary to accomplish this goal. However, I do not believe that this accomplishment is possible without the *active participation* of the student.

I have never considered education to be a passive activity. Students should be challenged to work hard, expected to excel, encouraged to ask questions, informed of their responsibilities, and rewarded according to their accomplishments. I demand active participation in the classroom and almost always conduct my lectures in "group-work mode", where I present new ideas (as a formal lecture), then have my students work together in small groups to discover the details by applying the ideas from the lecture to new problems. I firmly believe that my students actually learn more while *discovering* new concepts than they would were I to simply present the new concepts in a purely lecture format. This certainly does not mean that I do not lecture – I *always* strive to present clear, informative, structured lessons. An integral component of this is that there always be something beyond that which I present – something for the students to discover for themselves. I adopted this teaching philosophy very many years ago when I grew weary of the glazed-over look in the eyes of my students as I presented algorithms and ideas practically non-stop during the class period.

One of my biggest fears when I adopted this new teaching style was that I would not be able to cover all of the required material during the course of the semester. This has certainly not been the case – in fact, I am confident that my students are learning *more* under this system than they were before. I base this on their grades in the course, as well as their performance in subsequent courses. One rather subtle result of the group-work environment is that when students refer to previous material, it is most often in terms of "… this is sort of like when **we learned** such and such …" as opposed to "… this is sort of like when **you taught us** such and such …" The students themselves might not even be aware of the difference between these two comments, but faculty members certainly do!

Groupwork forms the foundation through which we have introduced additional goals in our courses. Although these goals are rather modest in our introductory level courses, they lay the foundation for much more lofty goals at the upper-level. Overall, these goals aim to better prepare our students to evaluate their own work as well as the work of others, to learn to work well with various groups, to hone their writing and presentation skills (while keeping their audience in mind), and to learn to research and master new material on their own.

2 Additional Goals for 1000- and 2000-level courses

A typical class meeting for 1000-level courses generally begins with me telling the students to find a new partner to work with today. (I insist that students work with different students for two main reasons: so that they learn to work with different people and so that students actually meet every student in the class. I believe that the latter helps build unity in the class.) I'll introduce the goal for the class period (for example, "Today we're going to learn the following sorts" or "We want to master multi-dimensional arrays before lunch"). I'll introduce the material and go through a few examples with the class. Then I'll give the class a group assignment and make my way from group to group as they work. As a group completes the assignment, I'll assign it new tasks. This allows the students who have learned the new material to continue to be engaged during class while I help those groups who are still working on the initial problem. I move from group to group, reminding the students "Be sure you are working **with** your partner and not **next** to your partner" – a mantra they hear often. Even on those occasions where we might spend the entire class period in "lecture mode", students are expected to contribute to the class discussion. (My least favorite class periods are those in which I'm the one doing most of the talking, especially if I'm up there in front of the class the entire time, instead of moving from group to group.) I take it as a sign of the success of this type of active-learning environment that students sometimes complain if we **don't** work in groups during a class period.

Beginning in our CS2 course, my students are expected to be able to read new material, work through it together with their partner(s), and then explain to me (or to another group) before the end of the class period. In fact, by the end of the semester, the students routinely work in groups to tackle rather difficult material, with the only "lecture" beforehand being about the topic in general. For example, toward the end of the semester, we cover Dijkstra's Single-Source Shortest Path algorithm (which many CS2 students find to be quite challenging at first look). I'll usually present the students with a network and have them try to determine the shortest path from some node to some other node. Then I'll ask about how to get from the start node to every other node. After this discussion, I'll tell each group to look at Dijkstra's algorithm and have them explain it to each other, along with a trace. They'll be told to let me know when they're ready to explain it to me. But by that point in the semester, the students notice nothing unusual about the task, since they've been steadily tackling other problems in the same manner. By the time students take my 2000- and 3000-level courses, they are very much accustomed to this learning style. They understand the benefits of working with others and automatically form new groups with little or no prompting. They also recognize that the expectation will be that they can look at new material and do the necessary work to understand and implement it.

Students in CS2 are also introduced to writing for a particular audience. Almost every semester, we have the students in CS2 write users guides/tutorials for the environment we use in CS1. Currently, we use JEdit, so the CS2 students must develop a users guide that is appropriate for CS1 students. The guides that are submitted usually run the gamut from excellent to what-were-you-thinking-when-you-wrote-this. The nice part of this assignment is that students already know how to use JEdit, so they can concentrate on the clarity and correctness of their writing. It gives us a chance to discuss the importance of very clear writing and of keeping the audience in mind. It gives students a "flavor" of one of the kinds of writing computer scientists do.

I also require my students to work on at least one out-of-class group project per course, beginning with the very first course they take. The ability to work well within various groups is a skill that any student who graduates with a degree in Computer Science should possess. Granted, group projects present many challenges when it comes to giving individual grades. In the past, I used to simply give a grade to the project, and each student in a group received the same grade. This usually resulted in a long line outside my office door as students lined up to "explain what happened on the project" and why their grade should not be the same as their partners'. I have since developed an evaluation form that students complete at each milestone during the project, wherein they evaluate their own work, as well as the work of their partners. I now give the project a grade and then use these evaluations to determine individual grades. In fact, in the upper-level courses, the evaluations themselves are graded. It is vitally important that students learn evaluation skills before entering the workforce. Details about how I manage and assess group projects can be found in [1][2][3].

In addition to the obvious goals for each class (mastery of the material, etc.), each of my courses has an additional set of outcome expectations, based on the level of the course. My philosophy for the introductory course (CS1) is that every student should be able to successfully complete the course. There is nothing *hard* about any of the concepts presented in the course, but there is a lot of material to be learned. As the students master the constructs, I often take opportunities to remind them that there are often several correct ways to complete most tasks. My "additional" goal for CS1 is that students become accustomed to this fact, and are able to identify additional alternatives for most solutions. This leads into the goal for the second class all majors and minors take: CS2. In this class, we learn to evaluate code and compare different possible solutions to a task, with the hope that a "better" solution can be attained. It is in this class that students learn to evaluate not only the code written by their team, but also code written by other teams. A typical class period includes introducing a new data structure or class, defining the methods the class will need, then having the students write the code for those methods. For the more-challenging methods, I will have each team put their solution on the board at the same time. Then each team will "rotate" to the next board, and evaluate the other team's code. They'll use colored chalk to trace through the code and indicate any errors or omissions they might discover. After evaluating one or two other teams' code, each team returns to its original code, where the team will indicate whether or not it agrees with the suggestions that were written in colored chalk. This practice has the advantage of allowing each student to see how other students (or groups) completed the task, and allows students to determine whether the other teams found "better" solutions. We spend a large amount of time learning to evaluate the computational complexity of code in order to determine which big-Oh category best describes the solution. As we know, it is very easy to determine which solution is "better" if the solutions fall into different big-Oh categories. When the solutions fall into the **same** big-Oh category, detailed analysis is required to determine whether one is "better" than the other. Being able to do this analysis is critical for the success of the students. The most-common question CS2 students hear from me is "Which is better?". I tell them early (and often!) that the answer to that question is almost always "It depends!", after which I'll ask "On what?". By the end of CS2, students are expected to be able to easily discuss and defend the Big-Oh analysis of all of the code they write. The goal for this course is that students learn how to evaluate a variety of code, as well as recognize the fact that it is the **code** that is being evaluated, **not** the programmer!

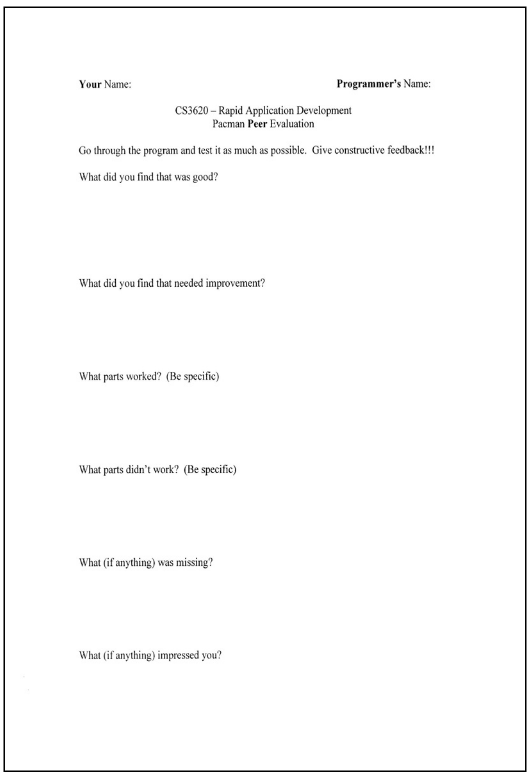
Goals for my 2000-level courses expand on this concept of evaluation. In Advanced Data Structures and Algorithm Analysis, we focus a lot of our attention on modifying existing code, often code that is found in the textbook. I have been fortunate enough to have found a book that actually includes many examples of code that contains a wide variety of shortcuts and "tricky" code– things which might often appear in code in the "real world". (I know that sounds strange, but it fits my purposes wonderfully.) Students will evaluate lots of the code in the book, attempting to figure out what the author was trying to do. After identifying the poorly-constructed coding features (my students learn very early on that I am a "Software Engineer" who does not usually approve of code that is not well structured!), the students are required to modify the code to meet "Schwartz standards". Modifying existing code is often the first job that most graduates will encounter. It is also a skill that is not always explicitly taught. Becoming comfortable in this area is the "additional" goal of this course.

3 Additional Goals for 3000- and 4000-level courses

I teach several upper-level courses, so most students have the opportunity to take more than one of their 3000- and 4000-level courses from me. I have found that the students generally consider the *first* upper-level course they take to be the "toughest", regardless of what course that happens to be. To address this, during the first class meeting of each semester, we talk as a group about the expectations I have of students taking upper-level classes. I let them know that I understand that things might occasionally *seem* overwhelming, but that they'll soon "get into the flow" of 3000-level work. I also remind them that if it seems like other students are more comfortable with the workload, those other students have likely *already taken* an upper-level course. I also remind the "new" students that they will be the ones others will look to for guidance next year.

For the past several years, our department has included a requirement in all 3000- and 4000-level courses that every project require some set of components that are not taught (or perhaps even discussed) in class. Students in upper-level courses must do additional outside (usually online) research to successfully complete each project. This might include finding out how to use additional user-interface features in Java, how to construct reports in Access and Oracle, mastering a new development environment (perhaps WAMP-based development or new transformation algorithms for 3-D graphics) or even finding, downloading and installing additional plug-ins for Eclipse. This feature sometimes is seen as unfair by students who are taking their first 3000-level course, but they quickly become accustomed to it and it just becomes a "natural" feature of upper-level course requirements. The fact that it is now a requirement for every project in every advanced-level course also makes it seem much less like a particular professor is somehow "punishing" students or not teaching them everything they need to know. (Of course, things work very nicely now that we have been doing it this way for a while, but the "transition" period was sometimes painful. A few of us took "hits" on our "student evaluation of instruction" numbers, but those hits were well worth the end result. In fact, it is now so much the "norm" that "experienced" students will usually explain to "newbies" (who are taking their first 3000-level course) that "that's just how it is". Our students actually take pride it being able to "teach themselves" as necessary.

My 3000-level courses include additional goals that move beyond working with code. I make it clear to students in these courses that the "starting point" for every program is that it adheres to the "best practices" covered in CS2 and Algorithms. So, when group projects are due in upper-level courses, the students use skills that are needed by project managers – evaluating team members (including themselves!), team cohesion and the extent to which their own projects meet the requirements stated in the specifications. This means that students evaluate how well each team member met his or her responsibilities. It also means that student groups must evaluate the level of correctness and completeness of their own project. This includes how aesthetically-pleasing their user interface is (when appropriate). Some students used to "balk" at having to explicitly state the flaws within their own projects, perhaps thinking that the grading process should include some level of "luck" – maybe the teacher won't notice that something is missing. To counter this, I explain to the students that they are not necessarily penalizing themselves with a thorough evaluation. I want to know ahead of time whether a program crashes under certain conditions so that I can save those test cases until after I have graded the rest of the project. I also tell them that I grade "extra hard" on upper-level projects, but will give them back up to half the penalized points on problem areas that they've included on their evaluation. The goal here is that students learn to assess not only the level of effort, but the level of success (or lack thereof) of every team member (including themselves). I often also have students evaluate the projects written by other groups (or individual students). For example, in my Rapid Application Development course (which is, basically, our Game Programming and User Interface Design course), on the day projects are due, students will use an evaluation sheet that I have developed to provide feedback on their own project, as well as feedback on up to three other students' projects (or however many projects that class time allows). A typical evaluation form is shown in Figure 1.


Your Name: **Programmer's** Name:

CS3620 – Rapid Application Development
Pacman **Peer** Evaluation

Go through the program and test it as much as possible. Give constructive feedback!!!

What did you find that was good?

What did you find that needed improvement?

What parts worked? (Be specific)

What parts didn't work? (Be specific)

What (if anything) was missing?

What (if anything) impressed you?

Figure 1. Sample Peer Evaluation Form

I keep the prompts very open-ended to allow for the widest feedback possible. I must admit that this entire process was not a well-researched one. To be honest, the first time I employed this type of evaluation was simply because I was "buried" under a pile of projects to grade and thought that I'd have the students do the "first pass" on the projects, so that I knew where to concentrate my efforts, grading-wise. I felt like a bit of a fraud and was sure that the students would see right through my scheme. However, I discovered I might just be on to something as I saw that the students took this task seriously and then started asking each other questions like "I was trying to get my Pacman to move more smoothly, but couldn't figure it out—how did you do it?" or "I really like how your user interface – how did you decide where to put everything?". I have since added a question&answer session after the evaluations are completed, so that students can freely exchange things they've learned while completing the projects. Just before students submit their sets of evaluations, I have them "rank" each of the projects, including their own, from 1st to 4th (or whatever), where the "best" project gets ranked 1st. This exercise allows the students to determine for themselves how their work "stacks up to" the work of their peers. Stronger students can "show off" their work without seeming arrogant or boastful, while weaker students (and, especially, slackers) know in advance that others will see their projects, which often provides extra incentive to get the work done. It has been my experience that students look forward to doing these evaluations, often believing these "eval days" to be a break from doing "actual work". It is often only after entering the workforce that students realize that they are quite proficient at evaluating others' work precisely because of these "eval days".

My Software Engineering course is by far the most "group-oriented" course. It is also our most writing-intensive course. During the semester, we tackle several very large projects, most of which are "service learning" projects in which we develop software for off-campus, non-profit organizations and schools. (Details about the these projects, along with the benefits of including service-learning projects, can be found in [4][5][6][7][8].) We follow the "standard" project development milestones (client interviews, requirements, specifications, preliminary design, detailed design, etc.). Client interviews for each project are carried out by the entire class – we usually have the clients come to class for these interviews. After all of the interviews are complete, random groups are assigned to develop Requirement Documents for the projects. At the end of this phase, each group submits a professional-quality document, presents its findings in class, and makes suggestions about what the next group should pay special attention to in the next phase. After these presentations are complete, different groups are formed and each group tackles the Specification document for a new project (one that no one in the group had worked on previously). This same process is continued throughout the design and development lifecycle of the projects. Research is also being conducted to the effect this process has on the overall quality of semester-long projects. [9][10] In addition to the obvious skills of learning to work with a variety of groups, students are also exposed to the "real life" experiences of producing high-quality documents at each phase of the development cycle and of jumping into the middle of a project and being expected to perform well.

4 Conclusions

As department chair, I have persuaded the other members of the department to adopt these additional goals whenever possible. I do insist that faculty teaching 1000-level courses require their students to identify alternate constructs (CS1) and to master the code-evaluation goals of CS2. However, each 2000- and 3000-level course is different, so I encourage faculty to attempt the goals whenever appropriate. I do insist that all upper-level projects for all classes include components that require outside-of-class research whenever possible.

We have noticed several results of our current approach to preparing our students for life after graduation. Our students are much more confident in their ability to successful job applicants and/or graduate students. We have an extremely high placement rate (within one month of graduation) in either fulltime jobs or graduate/professional/ law schools (usually with assistantships/fellowships) – I know of only one of our graduates for the past 8 or 9 years who didn't have such a placement. We also regularly get feedback from graduate schools and employers that our alumni are highly productive.

5 References

[1] Schwartz, Donald R., "Service Learning and Software Engineering: Managing 'Real World' Group Projects", *Proceedings of the 2007 International Conference on Software Engineering and Data Engineering (SEDE-07)*, Las Vegas, July 2007.

[2] Schwartz, Donald R., "Managing and Assessing Software Engineering Group Projects", *Proceedings of the 2005 American Society for Engineering Education Annual Conference and Exposition*, June 2005.

[3] Schwartz, Donald R., "The Benefits (and Drawbacks!) of "Real World" Software Engineering Semester Projects: A Case Study", *Proceedings of the 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-09)*, Las Vegas, July 2009.

[4] Schwartz, Donald R., "CLOWNS – A Software Engineering Semester Project Currently In-Use in Kindergarten Classes", *Proceedings of the 2008 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-08)*, Las Vegas, July 2008.

[5] Schwartz, Donald R., "Incorporating Service Learning into Capstone/Software Engineering Group Projects", *Proceedings of the 2010 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-10)*, Las Vegas, July 2010.

[6] Schwartz, Donald R., Jonathan F. Spencer, and Adam T. Huffman, "Service Learning, Software Engineering, and Hurricane Katrina: A Case Study", *Proceedings of the 2006 International Conference on Software Engineering Research and Practice (SERP '06)*, Las Vegas, June 2006.

[7] Kaye, Cathryn Berger. The Complete Guide to Service-Learning. Free Spirit Publishing. Minneapolis, MN. 2004.

[8] Underwood, Charles, Mara Welsh, Mary Gauvain, and Sharon Duffy. "Learning at the Edges: Challenges to the Sustainability of Service-Learning in Higher Education", UC Links. http://www.uclinks.org/reference/research/underetal_01.html .

[9] Anewalt, Karen. "Dynamic group management in a software projects course", Journal of Computing Sciences in Colleges, Volume 25 Issue 2, December 2009.

[10] Anewalt, Karen and Jennifer A. Polack-Wahl. "Teaching an iterative approach with rotating groups in an undergraduate software engineering course", Journal of Computing Sciences in Colleges, Volume 25 Issue 6, June 2010.

Probability and Statistics for Computer Science – Who Should Teach It?

Daniel Joyce
Department of Computing Sciences
Villanova University
Villanova, PA 19085

ABSTRACT
Despite initial reservations, after teaching a semester of Probability and Statistics for Computer Science, the author now firmly believes, for reasons described in this paper, that the course can be taught most effectively from within the computer science department.

Keywords: Probability, Statistics, Modeling, Active Learning, Assignments

1. Background

We are an accredited, medium-sized Computer Science (CS) program within a College of Liberal Arts and Sciences. For many years our majors were required to take a probability and statistics course offered by our Mathematics Department. This course, entitled "Statistics for Experimenters", was an excellent course - it had been created by the Mathematics Department for us in the late 1990s, and was taken by students from throughout the college, not just by computer science majors.

Last year a committee of members of our department recommended that we begin to teach the probability and statistics course ourselves, i.e., to teach it within our own department. The primary argument for the recommendation was that we would be able to use computer science related examples throughout the course, therefore allowing the students to appreciate the role of probability and statistics within computing. Such an approach would have been difficult for the mathematics instructors to accomplish, so the department accepted the recommendation.

During the discussion regarding the course, I was one of the few dissenting voices. I realized that the current course was sound and although I could see some benefit to being able to concentrate on examples from computing, I did not think that the status quo approach was "broken" - I did not think that the work needed to bring the course into our department would be worth the predicted benefits. As one of the few dissenting voices it was ironic that the second time we offered the course, I was slated to be the instructor.

This paper describes my experience teaching this course. It explains how my experience has changed my opinion. I now do believe that it is a good idea for us to offer the Probability and Statistics course from within our department, although my reasons are not exactly the same as the ones originally put forth by the committee.

2. Related work

It is easy to find areas of computing related to probability and statistics. Probabilistic algorithms use randomness, artificial intelligence learning patterns are often based on conditional probabilities, computer based modeling uses random number generation, software reliability analysis can use statistical regression techniques, and in general, many claims related to software usability and development can be studied using experimental procedures.

Some computer science educators have long called for an increase in the emphasis on "empirical methods" across the CS curriculum [1]. One might ask what should be taught to CS students in this area? A solid list of "Core empirical competencies for computer science" are listed in [2]. These include a basic understanding of probability, distributions, sampling, statistics, and experimental design. A recent paper on CS curriculum design [3] suggests that the core of CS be broken into six courses, with one of the six courses being "Probability Theory for Computer Science". This paper also argues that the correct place for a probability course for CS majors is within the CS department.

Papers describing courses similar to the one discussed in our paper have appeared. Notably, Anderson [4] describes a course on simulation, probability, and statistics taught within the CS department and fulfilling a college-wide "quantitative reasoning" requirement. More recently, Sahami [5] presents a course on probability theory for computer scientists, which is much like the course we describe in this paper.

Before leaving this section we should mention that textbooks targeted towards this course do exist and in our opinion, more will be available in upcoming years. Examples include [6], [7], and [8]. The Horgan book nicely incorporates the open source statistical processing environment R throughout the text.

3. Our Course

We have now offered this course twice, once in the spring of 2010, and once in the spring of 2011. This paper discusses the latter offering.

The course prerequisite is our first semester introductory programming course. It might be a good idea to raise this prerequisite to two semesters of programming, essentially requiring data structures, although in our environment scheduling concerns make this problematic. In any case, in our view, to reap the benefit of teaching this course within the CS department, it is essential that all students have some programming experience.

There were twenty-three students in the course, mostly second year students but with a few first year students and a few third year students. The class met three times a week in a lecture classroom environment. The course web site is

www.csc.villanova.edu/~joyce/csc5930stat/index.html

We had weekly quizzes, three tests plus a final, and multiple projects. The approximate schedule of topics was:

- Basic Probability
- Combinatorics
- Conditional Probability
- Families of Discrete Distributions
- Families of Continuous Distributions
- Descriptive Statistics
- The Central Limit Theorem
- Confidence Intervals
- Hypothesis testing
- Experimental Design

4. The use of Computer Science related examples

As mentioned previously, one of the primary reasons for bringing this course into the direct jurisdiction of the CS department was to allow the use of CS related examples. Throughout the textbook CS related examples are used, and almost all of the exercises involve CS. In many cases however the examples are strained. Let's consider a simple example:

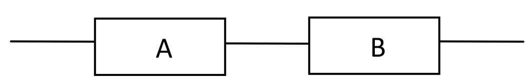
In a shipment of 200 computer chips, 3 are defective. If you choose two chips to use for a project, what is the probability that your project will fail due to a defective chip?

Such an example is obviously contrived. It is not really CS related. It is just a "computerized" version of a similar generic problem that would use the term "widgets" instead of "computer chips", or that would simply have a bag full of 197 blue balls and 3 red balls.

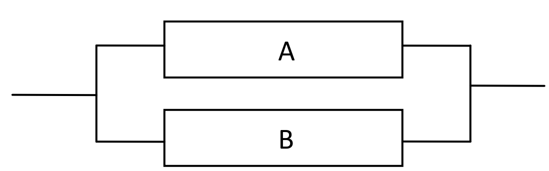
There is little pedagogic benefit to stating problems such as the example above using computer terminology. The students realize that it is not really related to computing. The additional verbiage required in such examples makes the example more difficult to follow and detracts from the real point of the example. It is our opinion that textbook authors and instructors would benefit from not trying to shove everything into a computer science context. It is pedagogically sounder to keep examples simple and uncluttered, such as by using the classic "balls in a bag" or by using the familiar pair of dice or deck of cards.

That said however, there are some places where good, solid, CS related examples can be used:

- Some of the subtleties of combining probabilities can be studied be considering the reliability of networks, based on the reliability of their components. If component A has reliability 0.9 and component B has reliability 0.8, then the network



has reliability 0.9×0.8 whereas the network



has reliability $1 - (1 - 0.9) \times (1 - 0.8)$.

- Conditional/Bayesian probability is often used as the basis for spam filtering approaches. Discussing and researching spam filtering offers a concrete basis for the study of these topics.

- Software reliability models are usually based on collecting historical data and applying regression analysis. This direct use of an "advanced" statistical approach within a software engineering context also allows CS students to appreciate the practical power of the theory.

- In addition to covering basic probability and introducing statistics, our course teaches the students that there is a place within computer science for the application of the scientific method [9]. A late semester project involves the selection by groups of a research paper related to CS that involves experimentation, and a short presentation by the group to the class describing the experiment, the dependent and independent variables, the statistics used, the internal and external validity, and the significance of the work.

So, it is not in the simple examples that we procure benefits by using CS related examples in our class, but rather in more complex examples and the ability to point out direct benefits of the material under study, helping to answer that age-old student question "why do we have to learn this?'

5. Programming

Opportunities abound within this course for creating programs that allow us to obtain insight into the material. During this past semester we created the following programs, among others:

- A program that “flips a coin” N times and prints out the ratio (showing 5 decimal places) of times Heads appears. It does this for N = 1, 10, 100, 1000, 10000, 1000000, and 10000000. This program clearly demonstrates the "Law of Large Numbers" with the ration approaching the expected value as more and more coins are "flipped" as shown by the following sample output:

```
ratio with 1 flips:       1.00000
ratio with 10 flips:      0.40000
ratio with 100 flips:     0.48000
ratio with 1000 flips:    0.47800
ratio with 10000 flips:   0.50640
ratio with 100000 flips:  0.49884
ratio with 1000000 flips: 0.49994
```

- A program that “flips a coin” N times and prints out the length of the longest "run" of heads or tails, again for N = 1, 10, 100, 1000, 10000, 1000000, and 10000000. Our intuition tells us that we could "never" flip, say, 20 heads in a row, but probability theory tells us that if we flip a coin enough times it will happen. This program verifies the theory (and helps give us confidence in our random number generator). Sample output:

```
# of Flips   Max Run
         1         1
        10         3
       100         7
      1000         8
     10000        13
    100000        19
   1000000        21
  10000000        22
```

- A program that simulates betting $10 on the Field of a craps table, over and over, 1,000,000 times. This is another example about the idea of expected value and teaches us not to bet against the house!

- A program that investigates the famous "Monty Hall" problem [10]. The interesting thing about this counter intuitive "puzzle" is that just the mere act of writing a program to simulate it clarifies the apparent paradox in the programmer's mind.

- A program that estimates the value of Pi by randomly generating points in a square and counting how many of them fall inside an inscribed circle (Pi is approximately equal to four times the number of points that land inside the circle divided by the total number of points.) This fun and surprising result demonstrates that the use of random number generation and probability can be used in interesting innovative ways.

- A program that generates a sequence of random numbers based upon an exponential distribution for a given λ, by generating random real numbers in the range 0 to 1 and then using the inverse of the cumulative distribution function. This program, which includes visual output in the form of a histogram, demonstrates the relationship between the probability density function and cumulative distribution function and provides a useful approach that can be used in Monte Carlo modeling. Sample output:

```
* equals 2265 occurrences

0   ***************************************
1   **********************************
2   ******************************
3   ************************
4   ******************
5   *************
6   **********
7   ********
8   ******
9   ****
10  ***
11  **
12  **
13  **
14  *
15  *
16  *
```

- A program that generates a multitude of samples from a given distribution in an attempt to estimate the (known) mean of the underlying population. For each sample the program calculates a 90% confidence interval, and then reports the results. From this project we learn, in an active hands-on way, what it means to call something a 90% confidence interval. This exercise strikes at the very heart of statistics.

In addition to the programs we created ourselves, either as exercises or those provided by the instructor, we used many example applets that are available on the web. We believe the fact that we programmed many examples ourselves made the use of such applets more "real" than if we had not experienced similar programming.

6. Bonus benefits of programming

In addition to the direct benefits of the programming problems and solutions highlighted in the previous section, we noticed several additional benefits to this approach. First, the programming assignments allowed the students to use their programming skills outside a programming class. It is nice for them to see that there can be applications for programs that are useful in classes besides their programming class. Furthermore, as we've written elsewhere [11], we believe there are many inherent benefits in simply having students program, especially for those students with a weaker background. More programming gives such students a chance to "catch up". In all cases where students were assigned a programming project they were later able to see and learn about an "expert" solution, thus enhancing their own approaches.

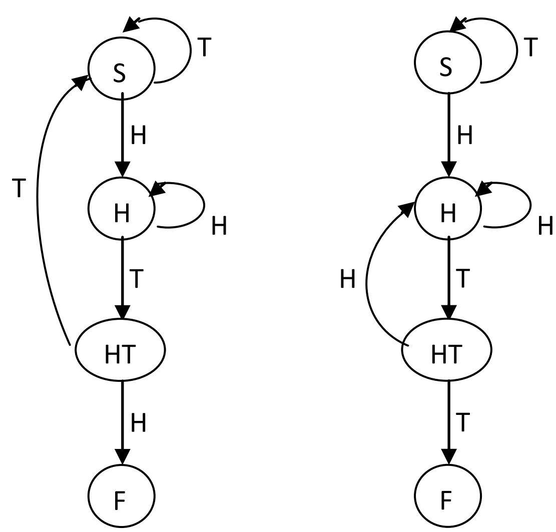
One assignment in particular led to a rather nice "bonus" benefit. We called this the HTX problem:

> Write a program that simulates flipping a coin over and over again until the sequence HTH has been seen, counting the number of flips required. This "experiment" should be repeated one million times; then output the average number of flips needed. Next repeat the entire process, except this time flipping until the sequence HTT appears. Again output the average number of flips. Include code, output, observations and analysis with your report.

Most people believe the average number of flips expected to obtain the sequence HTH is the same as the expected number of flips needed to get HTT. However, that is not the case, and a correctly coded program will report that on average we require about 10 flips to see HTH and 8 flips to see HTT. Note that we are not asking about the probability of flipping a coin three times and getting one of these sequences – we are asking for the expected number of flips until we get the sequence.

The benefit from this assignment is that the difference between the two results can be easily explained using Finite State Automata (FSA). FSA are, of course, of core importance within computing, playing a role not only in theory, but also in language, system, and software design. If we construct an FSA for each of the two desired sequences and compare we can easily see why the results are different:

for HTH for HTT



State S is the start state, F is the finish state (success), H is the state of having "seen" an H, and HT is the state of having "seen" an HT in sequence. As can be easily seen in the figures, if you have flipped an HT and are looking for HTH and you "fail", i.e. you flip a T, you go all the way back to the start state. However, if you have flipped an HT and are looking for HTT and you "fail", i.e. you flip a H, you do not go all the way back to the start state, instead you go back to the state of having seen an H and therefore you have already made some progress. It is this difference, easily seen using the FSA's that explains the difference in expected values. This example allowed us to talk about the use of FSAs outside the theory class and to show how one can easily implement an FSA in a program.

The final bonus benefit of programming was earned from the semester long class project. We had the students, in groups, design and build a web based survey, with results presented visually and dynamically. The "bonus" part of this is it allowed us to expose the students to some web programming technologies. About one third of the students already had plenty of experience with these technologies but the rest of the students were very happy with the chance to learn. This project was broken into the following phases:

1. Basic html – create a web page related to probability and statistics, including a link to an interesting video, and a list of suggested survey questions.

2. Server side – create a server side script that that demonstrates the use of echo, a loop, a decision, and file input and output.

3. Database – create a client side form that collects information and sends it to a server side script which saves the information in a database. Create a server side script that displays the information in the database.

4. Visualization – update the display report so that it uses some sort of graphical visualization.

We believe that many of these students will go on, of their own volition, and study these topics in more detail, now that they have been provided an introduction. And yes, we know this is not directly related to probability and statistics – but the project did include content components related to those topics and, as we said, the rest was bonus!

7. Conclusion

The Probability and Statistics for Computer Science course does belong within the computer science department.

8. References

[1] Empirical Investigation throughout the CS Curriculum. David Reed, Craig Miller, and Grant Braught, SIGCSE 2000 3/00.

[2] Core Empirical Concepts and Skills for Computer Science. Grant Braught, Craig S. Miller, David Reed, SIGCSE 2004.

[3] Expanding the Frontiers of Computer Science: Designing a Curriculum to Reflect a Diverse Field. Mahran Sahami, Alex Aiken, Julie Zelenski, SIGCSE 2010.

[4] A Course on Simulation, Probability, and Statistics. Scott D. Anderson, SIGCSE 2007.

[5] A Course on Probability Theory for Computer Scientists. Mahran Sahami, SIGCSE 2011.

[6] Probability and Statistics for Computer Scientists. Michael Baron, Chapman and Hall/CRC, 2007.

[7] Probability with R – An Introduction with Computer Science Applications. Jane M. Horgan, Wiley, 2009.

[8] Probability and Statistics for Computer Science, James L. Johnson, Wiley, 2007.

[9] Experimental Computer Science: The Need for a Cultural Change, Dror G. Feitelson, White Paper, was available at www.cs.huji.ac.il/~feit/papers/exp05.pdf on March 31, 2011.

[10] A problem in probability, Steve Selvin, letter to the editor, American Statistician, August 1975.

[11] Dealing with Experience Imbalance in Introductory Computer Programming Courses, Daniel Joyce, FECS 2010, July 2010.

Creating a 2+2 Program With Technical Colleges – the Bachelor of Applied Science in Information Technology

Rebecca H. Rutherfoord
Southern Polytechnic State University
Marietta, GA 30060

Abstract

One of the tasks that most universities face is the problem of transfer credits for students coming from other schools. This is particularly problematic with students who attend accredited two year technical schools and receive an AAS or AAT degree. Our university has created an articulation agreement with technical two year colleges who offer Associate of Applied Science (AAS) degrees. This paper will examine one of the 2+2 programs in the articulation – the BASIT - Bachelor of Applied Science in Information Technology.

Keywords

Articulation Agreements, Transfer Credit, 2+2 Programs

1. Background

For many states, 2+ 2 programs are designed to accept students who have completed their first two years and earned an associates degree (AS), to transfer to four year institutions to complete their bachelors degree. Examples of such programs include Monroe Community College in Rochester, NY [6], the University of North Carolina with North Carolina Community College System [1], Metropolitan State College of Denver with Front Range Community and Community College of Aurora [7], and Richland Community College with Benedictine University, Eastern Illinois University, Franklin Uniersity, Greenville College, Millikin University, Robert Morris College, Southern Illinois University at Carbondale, and University of Illinois at Springfield. [8] These are typical examples of 2+2 programs. Such programs have been accepted in several states around the country.

In our state, there are two separate college/university systems. One is the University System of Georgia (USG) that encompasses two-year, four-year, and research universities offering Associate, Bachelor, Master and doctoral degrees. The other system is the Technical College System of Georgia (TCSG). The TCSG comprises schools offering diplomas, certificates, and AAS degrees.

If a student wishes to transfer from a two-year college within the USG to another USG college/university, the transfer of credits is relatively seamless, and is done on a course-by-course basis. However, if a student wishes to transfer from a two-year TCSG college to a USG school, they are very likely to "lose" most of their degree credits in transfer. The AAT and AAS degrees of the TCSG schools do not contain many hours of general education. USG bachelor programs, on the other hand, have approximately 1/3 of credits in the general education area. In addition, most technical courses offered at TCSG schools do not have an equivalent course at the USG level.

Students are very frustrated that coursework taken at a TCSG college will most likely not transfer to a USG school. This means that a student, who wishes to continue his/her education beyond the AAS degree, must complete more than two years worth of courses to get a bachelor's degree. In fact, most students are required to take at least three years worth of additional coursework to get a bachelor's degree. This has caused many students to stop their education at the associate degree level.

Because of this problem, Southern Polytechnic decided that we would take a different approach that would allow students completing their AAS degree at a technical college to transfer to SPSU with the most credits possible. But SPSU has to address the normal problems with transfer credit.

2. Difficulties With Course Transfer

Many four-year institutions do not accept credit from AAS or AAT degrees. Four-year institutions often believe that the AAS or AAT degree courses are not rigorous enough to map into a four-year course curriculum.

The American Association of Collegiate Registrars and Admissions Officers, the American Council on Education (CHEA), and the Council for Higher Education Accreditation have created a joint statement on the transfer and award of credit. [4] They state that transfer of credit from one institution to another involves at least three considerations:

1. The educational quality of the learning experience which the student transfers;
2. The comparability of the nature, content, and level of the learning experience to that offered by the receiving institution; and
3. The appropriateness and applicability of the learning experience to the programs offered by the receiving institution, in light of the student's educational goals.

In addition, CHEA has a formal process of recognition which requires that all accrediting bodies so recognized must meet the same standards. These standards help to assure institutions that "accredited" schools that have accreditations recognized by CHEA can assume appropriate rigor and appropriateness of courses for transferability.

There is an additional problem, specific to for-profit institutions, most of which are not accredited by one of the standard regional accrediting bodies. Many of the for-profit institutions are accredited by the Accrediting Council for Independent Colleges and Schools. However, many schools and public institutions disregard this rating, and do not accept transfer credits from these institutions. Many students find themselves in the difficulty of having none of their credits transferring when they have attended a for-profit school. There are even law firms that specialize in this problem. Van Wey & Johnson, Attorneys at Law, believe that many for-profit institutions actually commit a type of fraud by not letting their students know that their credits will not transfer to most public or private accredited institutions. [9]

2.1 Regional Accreditation

Regional accrediting bodies, such as SACS – Southern Association for College and Schools, and NEASC – New England Association of Schools and Colleges, have position statements concerning transfer credit. [2,3] These regional accrediting bodies are recognized by CHEA. In its position statement, The Commission on Colleges of SACS states "Many systems and institutions have taken positive action such as negotiating articulation agreements…These kinds of proactive approaches, involving qualified faculty in the decisions, ease the way toward resolving transfer of credit problems while maintaining curricular coherence and academic and institutional integrity." The Commission on Institutions of Higher Education of NEASC states that "…policies and procedures should provide maximum consideration for the individual student who has changed institutions or objectives." [2,3]

One of the ways to help ensure quality in the transfer of courses is to examine the curriculum from the transfer institution, along with faculty and other pertinent accreditation criteria. Another is to accept only courses from accredited institutions. Since it takes a great deal of time to individually ensure quality from many different institutions, most accredited public, private and for-profit institutions have chosen to only accept credits from regionally CHEA recognized accredited institutions.

The state of Georgia has taken transfer credit between public institutions very seriously. It has created the Georgia College 411 website to help students plan, apply and pay for college in the state. It also has policies relating to transfer credits. [5]

At Southern Polytechnic State University (SPSU) in Marietta, GA, we have chosen to look at accepting transfer students from accredited institutions from the Technical College System of Georgia. We accept the complete AAS or AAT degree in a computing field. In order to better serve the students of Georgia, we have established an articulation agreement with most of the institutions within the TCSG. The articulation agreement was necessary so that both the University System of GA, Southern Polytechnic and the Technical College System of GA would all agree upon the courses that would transfer as part of an AAS or AAT degree.

3. Articulation Agreement

SPSU has entered into an articulation agreement with the Technical College System of Georgia in five different degree areas. For reasons that we explain below, the articulation is not into our regular bachelor of science programs, but into a different type of degree, namely the bachelor of applied science.

Our department has been a leader in establishing this articulation since we had already established a stand-alone articulation with one technical college located close to our own university. We have developed a BASIT – Bachelor of Applied Science in Information Technology degree program that fully accepts the two year AAS or AAT degree credits in a computing field from any regionally accredited TCSG school. The students then complete the last two years at SPSU to receive their bachelor of applied science degree.

The articulation agreement states that "…the transition from a SACS-COC accredited TCSG technical college to SPSU will be as smooth as possible for the students choosing to do so." The agreement goes on to state "…equal partners with the responsibility to maintain the integrity of their separate programs, to remain true to their institutional missions …" The agreement states that students must complete the associates degree at their respective technical college, and that sub-agreements will be completed for specific degrees.

The colleges in the Technical College System must be regionally accredited by SACS in order for SPSU to accept the transfer credits. The Chancellor of the University System of GA, President of the Technical College System of GA, and the President of SPSU all signed the official articulation agreement. The behind the scenes work on this articulation agreement was spearheaded by the Associate Vice President for Academic Affairs at SPSU who helped coordinate the creation of the articulation agreement. This took several months of negotiations between the two systems.

The articulation agreement is a strong document that spells out the expectations of both systems, and of Southern Polytechnic. It is very clear on how credits will be transferred and assessments of the agreement made.

The articulation agreement also determines primary contact representatives from the participating schools. It also covers the assessment process for the agreement by providing for a regular review of sub-agreements, at a minimum of once every three years.

4. Course Of Study for the BASIT

Students transferring into the BASIT program are required to complete their last two years – 60+ semester hours, at SPSU. The SPSU courses cover both general education and upper level IT courses.

As with our regular 4 year degree program – the Bachelor of Science in Information Technology, we want our BASIT students to obtain both breadth and depth. This is driven both by the demands of employers of our graduates, the ABET CAC accreditation standards in IT, and the IT model curriculum.

In line with 4 year programs offered at most other institutions, in our regular BSIT program, students usually obtain the breadth early in their academic program with courses taken in the junior and senior year providing depth. However, the structure of our BASIT program is different because students who enter the program have typically studied a particular area within IT in depth. For example, students may enter with an AAS or AAT in networking or web development. They thus enter SPSU with the required depth in IT but without the required breadth. The IT courses that they finish at SPSU therefore aim to provide a broad coverage of areas described in the BET CAC accreditation criteria for IT and in the IT Model Curriculum.

The fact that students transferring from TCSG schools obtain depth before they obtain breadth, and the consequent need to re-structure the courses that make up the degree program, is the primary reason for articulating into a BAS IT program, rather than into our regular BS IT program

In the area of IT, students take courses in

1. Programming (two courses)
2. Hardware/software concepts
3. Software acquisition and project management
4. Operating systems
5. Introduction to Web development
6. Advanced Applications Development
7. Data Communications & Networks
8. Information Security
9. Professional Practices & Ethics
10. Database
11. Two directed electives (such as E-commerce, IT and the Law or Management Information Systems)

In addition, they complete usually around 8 general education courses, including pre-calculus and discrete math.

4.1 Delivery Mode

The fact that TCSG institutions, in common with community colleges and technical colleges elsewhere, have a narrowly defined regional mission influenced the delivery mode of the courses that make up the BASIT. All courses are offered in a variety of modalities, including completely on-line, face-to-face and in a blended or hybrid mode, with half the session online and the other face-to-face. The fact that we deliver courses fully online allows students from across the state to participate in the program without leaving their home town. This is very attractive to students who want to continue working in their hometown area. We have full fall and spring semesters and a shortened summer semester.

4.2 Orientation and Advising

SPSU has created an orientation for students coming into the BASIT program that is presented both as an on-campus version offered several times prior to a semester starting, and an on-line version for students who are not able to come to the campus orientation. This helps students acclimate into the program and be successful. Orientation covers the normal topics that one would expect to be covered in any orientation session, as well as a tutorial on the course management system that the students will use as they complete their classes and an introduction to the demands of online learning. We believe the latter to be especially important as many students enrolling in the BASIT program will be taking courses for the first time by distance online learning.

4.3 Reception by Students

There has been a very enthusiastic response to the BASIT program. Inquiries and applications have been steadily growing as more students from the technical colleges are made aware of the program. We are currently just shy of 100 majors in the program since this program started 3 years ago. We believe that we are serving the citizens of Georgia by providing a seamless 2+2 program tailored for the AAS and AAT accredited degrees. It would appear that this will be an extremely popular program for many years to come.

We believe that this type of program can be replicated at other institutions with careful consideration and cooperation between technical colleges and traditional four-year institutions. Clearly spelled out articulation agreements, and careful consideration of curriculum can lead to academically prepared students who have a very strong technical background in Information Technology. Programs such as ours can assist students who have been stymied with AAS and AAT degrees to pursue bachelors degrees as well without losing transfer credits. We encourage other institutions to consider such a program.

5. References

[1] http://online.northcarolina.edu/2plus2/

[2] New England Association of Schools & Colleges, Commission on Institutions of Higher Education. "Transfer and Award of Academic Credit", http://cihe.neasc.org/downloads/POLICIES/Pp75_Transfer_and_Award_of_Academic_Credit.pdf

[3] Southern Association of Colleges and Schools, Commission on Colleges. "Transfer of Academic Credit, A position Statement", http://www.sacscoc.org/pdf/transfer%20credit.pdf

[4] http://www.acenet.edu/Content/NavigationMenu/ProgramsServices/CLLL/Credit_Transfer.pdf

[5] http://www.gacollege411.org/AdultLearner/transfer/transferring_credit.asp

[6] http://www.monroecc.edu/depts/careercenter/StuServ/h-2+2Programs.htm

[7] http://www.mscd.edu/admissions/twoplustwo/

[8] http://www.richland.edu/academics/articulated

[9] http://www.vanweyjohnson.com/PracticeAreas/Accreditation-Credit-Transfer-Education-Fraud.asp

A Spatial Database Benchmark for Classroom Application

Weijun Huang, Philip J. Bernhard
Department of Computer Sciences, Florida Institute of Technology, Melbourne, Florida, 32901

Abstract - *In this paper we describe a spatial database benchmark for classroom application called Space Bench. In many ways this benchmark is similar in structure to the industry standard TPC-h benchmark. Space Bench, however, is designed to test the performance of a database system that supports spatial types and operations, as defined by the OpenGIS spatial specifications [1,2]. Specifically, the benchmark includes a spatial database schema, a collection of spatial queries, and a data generating program for generating spatial data. Initial results indicate that this benchmark is very helpful when used as a project in an advanced course on spatial database systems.*

Keywords: Spatial Database, Benchmarking, Computer Science Education

1 Introduction

It is very common in computer science database courses to require students to perform a variety of projects. Depending on the specific course, such projects take a variety of forms including developing Entity-Relationship (ER) diagrams, coding database schemas in Data Definition Language (DDL), coding and executing (SQL) queries, or writing application programs that operate on a database via ODBC, JDBC or some other API. Frequently a course project will involve a combination of such tasks.

Over the past decade we have had success using the industry standard TPC-h benchmark [3] as the basis for several course projects in our database courses, both at the graduate and undergraduate levels. Although not originally intended for classroom application, the benchmark presents students with a variety of challenges that are typical of real-world database projects. Challenges that we believe are absent from many traditional academic database projects.

Given our success with using the TPC-h benchmark in our database courses, we developed a spatial database benchmark for classroom application called *Space Bench*. In many ways this benchmark is similar in structure to the industry standard TPC-h benchmark. Space Bench, however, is designed to test the performance of a database system that supports spatial types and operations. The benchmark is intended for use in a special topics course, or an advanced course on spatial databases.

2 The TPC-h Benchmark

TPC-h is an industry standard decision support benchmark published by the Transaction Processing Performance Council (TPC) [3]. The benchmark has been around for over a decade, and has been used and accepted throughout the database industry. Results from application of the benchmark are publicly available on the TPC website [4], and are updated regularly as companies provide results. In addition to the TPC-h benchmark, TPC currently publishes other benchmarks, including benchmarks for on-line transaction processing (TPC-C, TPC-E), and energy consumption (TPC-Energy). Over the years, TPC has published other benchmarks, which they now refer to as obsolete. The reader is referred to the TPC website for additional details.

At a high-level, three primary components of the TPC-h benchmark are a database schema, a data generating program, and a collection of 22 SQL queries. The database schema is based on a parts/supplier database application. The database includes detailed information on parts, their suppliers, orders submitted by customers, individual line-items in those orders, plus nations and regions of the world where suppliers and customers are located. The schema is relatively straight-forward, and uses simple (non-spatial) SQL types, constraints, and primary and foreign keys.

The data generating program for the benchmark is coded in C, and capable of generating anywhere from small to very large quantities of data. Results currently published on the TPC website reference databases ranging in size from 100GB to 30TB. In the past, we have found this program very easy to use, and have used it to generate data sets as small as 1GB for classroom application.

The 22 SQL queries for the benchmark are non-trivial decision support queries that make use of many standard SQL features and capabilities. This includes nested queries, grouping, sorting, aggregate functions, multi-table joins, and character string matching.

3 Using TPC-h as a Classroom Project

The TPC-h benchmark can be used as the basis for a variety of different classroom projects. The primary way in which we have used it in our introductory level database class is as a capstone project where students are required to build a database, load it with data, and execute queries. Students are provided with a description of the TPC-h schema, a 5GB data set for the database, and the text of the 22 queries TPC-h benchmark queries.

Students are required to code a TPC-h DDL file based on the schema description, create a TPC-h database using a relational database management system (DBMS) of their choice (subject to a few restrictions), load that database with the provided data, and execute the 22 queries. Note that students are also required to tailor the 22 queries in some minor ways to accommodate DBMS SQL syntactic variations.

Additionally, we have also used the benchmark in our advanced database class, building on the database constructed in the introductory class. Specifically, students are required to load a larger amount of data (10GB), run experiments that gauge the efficiency of the queries, and then tune the system by building indexes, adjusting memory allocation, etc., in order to improve query performance.

4 A Spatial Benchmark

In addition to the two database courses mentioned above, we also teach an advanced special topics course on spatial database technology. Broadly speaking, this class focuses on the impact that spatial concepts have on all aspects of relational database technology, including data modeling, database design, query writing, and DBMS implementation. Typically the course is only taught at the graduate level, and students are expected to have completed the two courses mentioned above as prerequisites.

In many ways this course covers the same material as the other two courses, but with a specific focus on how the traditional database technologies are extended to accommodate spatial concepts. For example, ER-diagramming is re-taught to the students, but with specific emphasis on how spatial concepts are modeled. Similarly, students are taught to write queries in Structured Query Language (SQL), but with a spatially extended version of SQL consistent with the OGIS spatial standard [2], which includes a rich set of spatial types and operations.

Our success with using the TPC-h benchmark in the first two courses led us to conclude that it would be helpful to have a TPC-style benchmark focused on spatial types, data and queries for use in our spatial database class. As with the TPC-h benchmark, such a benchmark could be used as the basis of a variety of projects. Students could use it to build a spatial database, generate and load it with spatial data, and execute spatial queries. Additionally, students could also use it to explore performance issues related to spatial indexing and memory allocation.

To the best of our knowledge there is currently no such benchmark available, either on the TPC website or anywhere else, so we set out to develop one. The resulting benchmark, called *Space Bench*, is similar to the TPC-h benchmark in that it consists of a spatial database schema, a collection of spatial queries, and a spatial data generating program.

Our current work is focused primarily on the data generating program, which is capable of generating a variety of spatial data types including points, line-strings, and polygons. For example, Figure 1 shows a small sample of data generated by the program. This particular figure contains three types of spatial data– points, triangles, and random-walk line-strings.

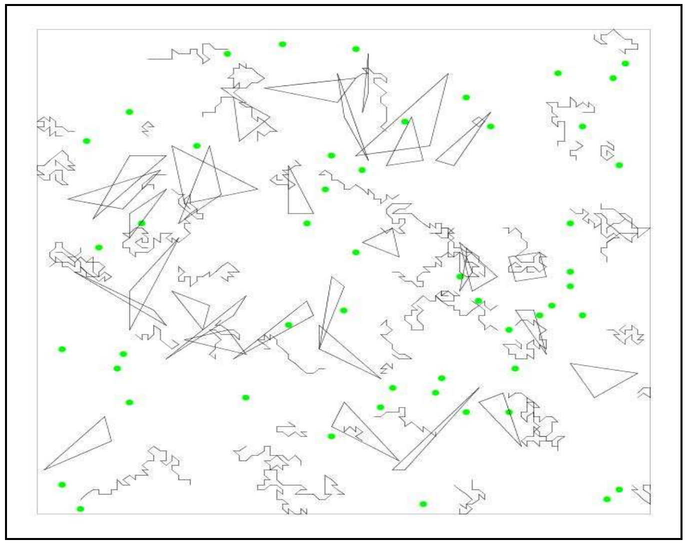


Figure 1: A sample of data generated by the Space Bench data generating program.

At a high level queries in the benchmark are similar to SQL queries, in that they have a *select-from-where* structure. They differ, however, in that they consist almost exclusively of spatial operations applied to spatial types. For example, Figure 2 shows a query that identifies the triangle(s) having the maximum number of line intersections within 300 units. This particular query makes use of two tables – one containing data on triangles (the *triangle* table), and one containing data on line-strings (the *line* table).

One important decision we faced during the early stages of query and schema design for the benchmark was the extent to which spatial types and operations would be mixed in queries with more traditional (non-spatial) SQL types and operations. Many traditional SQL types and operations can have a significant negative impact on query performance, including grouping, sorting, and non-spatial joins, and we wanted students to experience the full impact of spatial types and operations on database performance, without interference from types and operations that were not spatial in nature. Hence, non-spatial, performance intensive types and operations have been minimized, and in some cases completely excluded from Space Bench queries. For example, the query in Figure 2 focuses almost exclusively on spatial types and operations.

Similarly, the Space Bench database schema is based primarily on spatial types. More specifically, the Space Bench database schema has tables for points, line-strings, and polygons, where each table contains an identifier that functions as the primary key for the table, plus a spatial attribute called *shape*, that contains appropriate spatial information.

```
CREATE VIEW IntersectingLines AS
SELECT line1.id AS lineA, line2.id AS lineB
FROM line as line1, line as line2
WHERE Intersect(line1.shape, line2.shape) = 1

CREATE VIEW nearLineIntersect AS
SELECT triangle.id, Count(IntersectingLines.lineA) AS
num_intersections
FROM triangle, IntersectingLines, line as line1, line as line2
WHERE
IntersectingLines.lineA = line1.id AND
IntersectingLines.lineB = line2.id AND
Intersect(Buffer(Center(triangle.shape), 300),
Intersection(line1.shape, line2.shape)) = 1

SELECT triangle.id, num_intersections
FROM nearLineIntersect
WHERE num_intersections = ( SELECT
Max(num_intersections) FROM nearLineIntersect )
```

Figure 2: A sample Space Bench query.

5 Conclusions

To date we have used Space Bench as the basis for a course project in one instance of our graduate level spatial database course, and we expect to use it again in the near future. Initial results indicate that this benchmark is very helpful when used as a project in an advanced course on spatial database systems. As with the TPC-h benchmark, Space Bench can be used as the basis for a variety of different projects involving data modeling, design, implementation, and performance analysis.

6 References

[1] "*OpenGIS Implementation Standard for Geographic Information – Simple Feature Access – Part 1: Common Architecture*," Open GIS Consortium, August, 2010.

[2] "*OpenGIS Implementation Specification for Geographic Information – Simple Feature Access – Part 1: Common Architecture & Part 2: SQL Option*," Open GIS Consortium, August, 2010.

[3] "*TPC Benchmark H (Decision Support) Standard Specification,*" Transaction Processing Performance Council, 1993.

[4] The Transaction Processing Performance Council, *www.tpc.org*.

SESSION

RECRUITMENT AND RETENTION METHODS + ASSESSMENT + ACCOUNTABILITY STRATEGIES + RELATED ISSUES

Chair(s)

TBA

The Tricked-Out Virtual Body Shop: Recruiting African-American High-School Students to STEM through Automotive Design

Mark Howard, A. Nicki Washington, Todd Shurn, Legand Burge, Grant Warner
Department of Systems and Computer Science and Department of Mechanical Engineering
Howard University, Washington, DC

ABSTRACT

The Tricked-Out Virtual Body Shop is an extrapolation of engineering curriculum design activities that have successfully engaged African-American students. Declining enrollment among underrepresented African-American students in STEM may be related to negative perceptions of science, technology, engineering, and mathematics (STEM). This concept is designed to directly engage students in constructing a customized virtual automobile, while simultaneously educating them to the science and engineering behind the design and development process.

Keywords
Virtual Worlds, Science Technology Engineering and Mathematics, E-Learning, Multi-User Virtual Environments. Virtual Reality.

1. INTRODUCTION

The United States has experienced a steady decline in the enrollment of underrepresented minority students in science, technology, engineering, and mathematics (STEM) disciplines. In addition, the retention and graduation rates of this demographic have also declined [10]. As the baby-boomer generation enters retirement, a growing gap, known as the “Quiet Crisis,” describes the increasing demand for and limited availability of qualified U.S. citizens available to replace them [10]. This fact, coupled with the changing demographics of the American society, makes STEM education an area requiring urgent national attention.

The minority population is projected to increase at a much faster rate than the non-minority population over the next 50 years. The African-American and Hispanic workforce population will increase from a combined 22% to 40% of the U.S. workforce [9]. In order to quell this crisis, a large increase in students, specifically underrepresented, must enter and remain in the STEM pipeline from K-16. How to best attract and retain these students, who traditionally underperform in science and math, is a critical question.

The declining enrollment and retention of underrepresented students in STEM is partly related to their negative perceptions [3]. Students express a lack of appreciation for the relevance and applicability to life, finding them boring and irrelevant [4]. In addition, many minority students lack the strong math and science background required to successfully pursue a degree in these fields [8] [10].

To improve the perception of STEM to this demographic, we developed The Tricked-Out Virtual Body Shop, a context-based virtual design environment that can leverage any multi-user virtual environment (MUVE). This concept has three goals. 1) change the perception of minority students, specifically African-American, to STEM, 2) increase the interest of this demographic to collegiate STEM programs, 3)serve as a blueprint for affinity-focused STEM programs seeking to attract and retain minority students through the utilization of context-based virtual learning environments.

The remainder of this paper is organized as follows. First, we discuss the motivation of this concept. Next, we discuss the methodology and approach to the development of the tool. Third, we discuss potential implementation. Fourth, we discuss the evaluation of the tool. Finally, we conclude our findings.

2. MOTIVATION

There has been extensive research integrating e-learning systems and multi-user virtual environments (MUVEs). This new paradigm leverages the combination of interactive, 3-D games with an improved sense of presence and community not previously experienced in e-learning systems. In addition, knowledge building, sharing, and representational aspects of e-learning systems were greatly improved from prior implementations. Second Life Object-Oriented Dynamic Learning Environment (SLOODLE) is a popular MUVE application [2].

Many colleges and universities have created a Second Life presence that includes virtual on-campus classrooms and events. While more than 700 colleges and universities have a presence in Second Life, there is, to the best of our knowledge, only one HBCU, Alabama A&M, with a campus on Second Life. In an effort to attract and engage more African-American students to Second Life, Linden

Labs, the producers of Second Life created HBCU World, a virtual campus designed to expose students to HBCUs across the U.S., including virtual campus tours, and information on preparing for college.

While this effort has been launched, there is still a large, open area of opportunity for engaging African-American students in STEM through MUVEs. The Tricked-Out Virtual Body Shop is a unique concept that will provide an innovative experience to an untapped population of students. It is our goal that any university interested in STEM outreach activities involving underrepresented students, particularly African-American and Hispanic-Americans, can leverage The Tricked-Out Virtual Body Shop as a blueprint for their activities, with implementation in a number of MUVEs.

This concept was designed to utilize the urban/hip-hop culture to attract and engage underrepresented minority students to STEM. Our automotive theme leverages a popular activity among students within this target age group known as “tricking out” a a car: customizing cars with various decorative and mechanical features. Many students are familiar with this idea from music videos and television shows, such as MTV’s “Pimp My Ride.” On this show, cars are “tricked out” to contain unique custom features, including hydraulics, engines, audio systems, paint, and more.

This concept is the basis for the Tricked-Out virtual Body Shop. Students visiting the virtual shop will choose a vehicle type, and customize it to operate on different power trains. They would also be able to add additional designs and features, including their own custom creations. Students would then be able to test their custom vehicles against others in competitions. By comparing designs and results, students would ultimately learn more about mechanical engineering and computer science concepts and real-world application.

We note that, while we consider mechanical engineering and computer science as the disciplines of focus for this project, there are a number of other disciplines that can be incorporated into this activity.

3. METHODOLOGY

The virtual automotive design paradigm is based on the “Three C’s”: community, content, and context.

Community

The community is the environment that supports the virtual users and all in-world interactions and activities. It is simply the MUVE or virtual world that hosts a virtual learning session or environment. The Tricked-Out Virtual Body Shop will provide a virtual 3-D, immersive education environment, where students can interact with each other and instructors, while engaging in active learning.

Content

Content, refers to the pedagogy, material artifacts, and concepts created and used to study a discipline. Design requirements and support materials will be supplied by computer science, mechanical engineering, and graphic arts departments. [7].

Context

Context refers to the representation of real-world active learning objects and activities, using thematic references to establish meaning and relevance to the content and community supported. For example, the developed theme, “Tricked-Out Virtual Body Shop,” teaches pre-college students about advanced concepts in mechanical engineering, such as drag, acceleration, and efficiency, while marrying these concepts to real-world problems such as carbon emissions reduction and the use of alternative fuels for efficiency and global warming reduction.

3.1 Approach

The floor plan of the virtual body shop is illustrated in Figure 1. The main floor plan consists of a common area for interaction with instructors and other students. Virtual automobiles are driven into the virtual garage where students have access to three staging areas focused on the following: 1) body work 2) power plant and 3) simulation. Students within the power plant or body work areas will be able to customize vehicles by adding components (i.e. power and fuel sources, and body work) using drag-and-drop tools. Figure 2 illustrates customization possibilities.

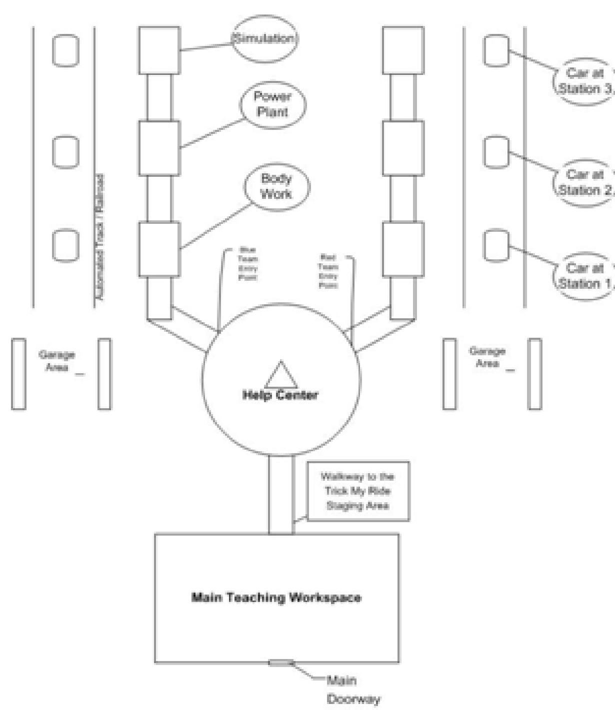


Figure 1. Tricked-Out Virtual Body Shop Floor Plan

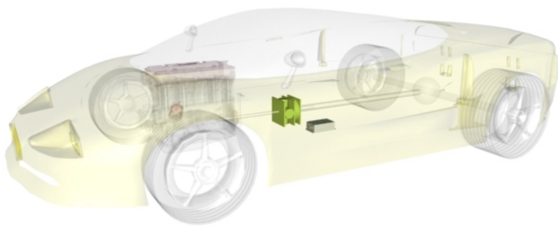


Figure 2. Body and Power Train Customization

The proposed environment serves two purposes. First, it functions as a learning and exploration workspace. Second, it serves as a virtual lab environment for competing users. The design must incorporate a shared area for learning and exploring, and then allow students to separate into groups, where they work collaboratively and competitively to further develop their vehicles. In the virtual lab environment, each group will be able to apply the principles learned in the exploration space to guide them in the development of their vehicles. Design feedback is accomplished by performing simulated runs on a dynamometer, such as the one in Figure 3.

Students can participate in a number of challenges, based on the intended learning objectives. For example, one challenge, to be held in the simulated staging area, provides each team with a base model care, fitted with a standard internal combustion engine (ICE). The car has an efficiency graph, similar to the one shown in Figure 4. Each team is responsible for tweaking the car design, so that the peak achievable efficiency is obtained, as in Figure 5.

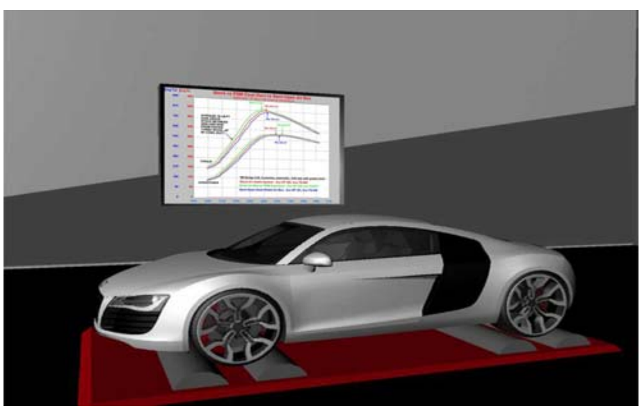


Figure 3. Simulated Dynamometer

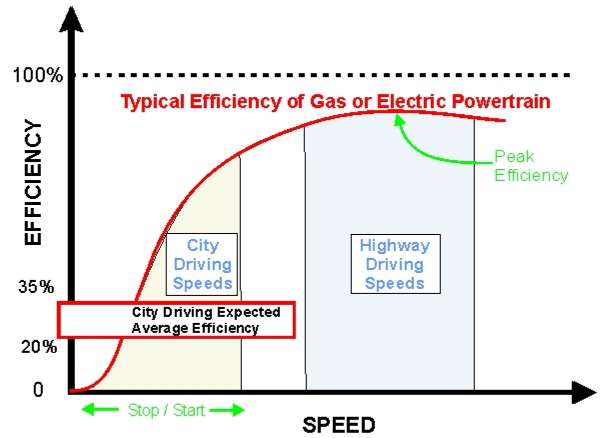


Figure 4. Gas or Electric Power Train Efficiency.

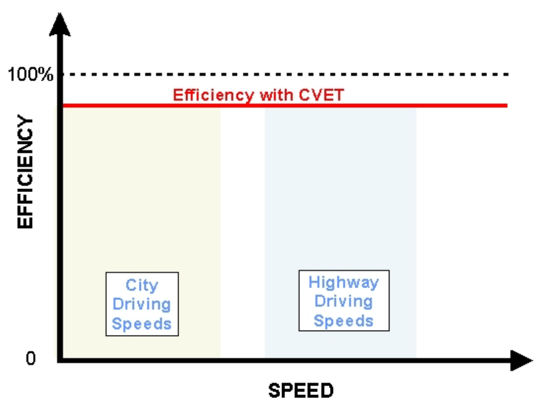


Figure 5. Peak Achievable Efficiency of the Gas or Electric (Hybrid or Fuel Cell) Motor with Continuous Variable Electronic Transmission (CVET).

Students complete this challenge by applying the knowledge of different types of power trains available for hybrid energy vehicles (HEVs), obtained in the learning and exploration workspace. To complete the challenge, students must determine which HEV technologies to use in their model car, and then perform simulated runs on the dynamometer, to view the resultant efficiency graph. The first team to complete the challenge within the allotted time, or have the most complete results, is the winner.

We are currently considering an additional stage, which would include a "Fast and Furious"-inspired race track. Student teams would be given equal amounts of fuel. The goal of the challenge would be to drive their custom car around the track and achieve the most possible revolutions.

4. Implementation

The Tricked-Out Virtual Body Shop can be implemented in a collaborative multi-user virtual world such as Open Wonderland. Open Wonderland is a good choice because it is open source, Java based and has minimal server requirements for hosting a virtual world. Creating the "Tricked-Out Virtual Body Shop" as a 3-D virtual world could be done as a collegiate computer science course project or an independent study. Many high-school students would be able to directly modify or extend the body shop environment, as Java is the primary high-school programming language.

The implementation of the Tricked-Out Virtual Body Shop should utilize built-in Open Wonderland object sharing, communication and virtual collaboration features. A direct link to the virtual laboratory (within the virtual world) for testing virtual automotive designs will provide impartial design assessment to individuals or group competitors. Aggregated results from virtual laboratory tests will be continually processed, to support immediate automotive design rankings.

The Open Wonderland client will support approximately 50-60 users on a typical server configuration. This allows a few small classes to use the same instance of the virtual environment. Scalability of the Tricked-Out Virtual Body Shop beyond a single class or school is necessary, to provide students design feedback across a large student population. Scaling may be accomplished by daisy-chain linking virtual laboratories. Body shops themselves do not require scaling, as only the automotive designs are competitively ranked. Ultimately, automobiles designed in the Tricked-Out Virtual Body Shop should be exportable to virtual worlds like Second Life, virtual learning environments like Moodle, and social networks like Facebook, for applicable awards.

5. Evaluation

The Tricked-Out Body Shop evaluation will use summative and formative methods to assess: (1) how pre-college students view STEM, (2) student interest in pursuing careers and college majors in STEM, and (3) student technical skills in STEM. The summative evaluation will assess the effectiveness of the program relative to the aforementioned goals, including quantitative and qualitative components. Goals 1 and 2 will be assessed by evaluating the results of the Strong Interest Inventory, which measures interest in general occupational themes, and interest, occupational, and personal style scales [8].

Reports of the Strong Interest Inventory provide substantial detail, including standard scores and qualitative descriptors for each scale. The evaluation hypothesis will compare the scores (related to STEM) of students participating in The Tricked-Out Virtual Body Shop to students from the same demographic that did not participate. Goal 3 will be evaluated using structured interviews of student participants, to access any changes in technical skill development, knowledge attainment, and general interest related to STEM.

6. CONCLUSION

In this paper, we present the Tricked-Out Virtual Body Shop concept, leveraging 3-D virtual worlds to attract and retain underrepresented minorities in STEM disciplines. By using automotive design, the presented work aims to add context-based relevance to STEM disciplines. By manipulating automobile components within the Virtual Body Shop, we seek to impact the perception of the target demographic and ultimately counteract the steady decline in enrollment and retention.

7. REFERENCES

[1] Harasim, L. Collaborating in Cyberspace: Using Computer Conferences as a Group Learning Environment, Interactive Learning Environments, Volume 3, Issue 2 1993 , pages 119 – 130.

[2] "Sloodle Website." January 29, 2009. Retrieved March 1, 2011 from http://slisweb.sjsu.edu/sl/index.php/Sloodle.

[3] Sarita Yardi. "What Is Computing? Bridging the Gap Between Teenagers' Perceptions and Graduate Students' Experiences", ICER 2007, 39-50.

[4] A. McGettrick et.al. "The current crisis in computing: what are the real issues?" SIGCSE Bull., 39(1), 2007.

[5] "Project Wonderland. Retrieved March 1, 2011 from http://en.wikipedia.org/wiki/Open_Wonderland.

[6] "Generation Q." Retrieved Feb. 20, 2011 from http://query.nytimes.com/gst/fullpage.html?res=9A07E2D71E3DF933A25753C1A9619C8B63.

[7] "Ford Stem Donation." Retrieved March 11, 2011 from http://mediagrid.org/news/2008-03_Ford_STEM_Donation.html.

[8] "Strong Interest Inventory." Retrieved March 11, 2011 from https://www.cpp.com/products/strong/index.aspx.

[9] Steward, R. et al. "Alienation and interactional styles in a predominantly White environment: A study of successful Black students,"Journal of College Student Development, 31, 509-515, 1990.

[10] National Academy of Engineers. "The Engineer of 2020 Project Prospectus," Retrieved March 20 from http://www.nae.edu/nae/engeducom.nsf.'

[11] Jackson, S.A. "The Quiet Crisis: Falling Short in Producing American Scientific and Technical Talent."Building Engineering and Science Talent. www.bestworkforce.org/ODFdocs/Quiet_Crisis.pdf.

Development of a Data Analysis Environment for Assessment

L. Onitsuka[1] **and M. J. Lee**[2]
[1]College of Engineering and Computer Science, California State University, Sacramento
Sacramento, CA, United States
[2]Computer Science Department, California State University, Sacramento
Sacramento, CA, United States

Abstract - *Assessment plays a vital role in the ability of an academic program to evaluate its educational objectives and student learning outcomes. The Computer Science Department determined that a system that provides an integrated environment supporting data analysis as well as linking program objectives and student outcomes would help the department organize, manage, and publicize its assessment efforts. This project is a web-based data analysis tool for assessment. The public can view non-sensitive assessment information, and the faculty can access not only public information but also any sensitive information. This includes interviews, evaluations, surveys, and other forms of raw data. In addition, faculty have access to methods which will be used to analyze the data, in particular, data resulting from the implementation of a specified rubric.*

Keywords: assessment, tool, web, data analysis, rubric

1 Introduction

Assessment plays a vital role in the ability of an academic program to evaluate its educational objectives and student learning outcomes. Making assessment information available to students, faculty, campus assessment coordinators, and the public is as important as publicizing its programs. The Computer Science (CSC) Department determined that a system that provides an integrated environment supporting data analysis as well as linking program objectives and student outcomes would help the department organize, manage, and publicize its assessment efforts. Not finding such a system, a prototype system was designed and developed [1].

The system was designed to have three levels of users: public, data entry, and faculty. The public can view non-sensitive, general assessment information. In addition to accessing public information, the data entry view provides the ability to insert, delete, or modify data. This includes preparing rubric forms and entering the raw data associated with those forms. The faculty view provides access not only to public information but also interviews, evaluations, surveys, industrial visits, and other forms of raw data gathered for assessment. In addition, faculty have access to a menu of available analytical methods. These methods will be used to analyze the data resulting from the implementation of a specified rubric.

2 Design

As a web-based application, the system is designed to provide an efficient and flexible environment for entering data and utilizing data analysis tools. It provides a means to effectively reach a large audience. The system uses a three-tier web application architecture. The presentation layer is the user interface available via the internet. The business layer is the PHP processing that sends and receives requests between the presentation layer and the third layer, the data layer. The data layer is a MySQL database. Overall, it utilizes a LAMP (Linux, Apache, MySQL, PHP) open source web platform. As a measure of security, only the owner and web daemon have access to the files.

At the top level, the mission statements for the University, the College of Engineering and Computer Science (ECS), and the CSC Department along with the program educational objectives are listed. The second level is the student learning outcomes. These are “narrower statements that describe what students are expected to know and be able to do by the time of graduation. These are related to the skills, knowledge, and behaviors that students acquire in their matriculation through the program [2]”. The relationship between objectives and outcomes are represented in a matrix. Individual objectives and outcomes can be assessed directly. The assessment tool allows a given outcome to be broken down into categories. These categories are then broken down into specific performance criteria. Each criterion is then assessed and evaluated. Figure 1 visually displays the goals of this system.

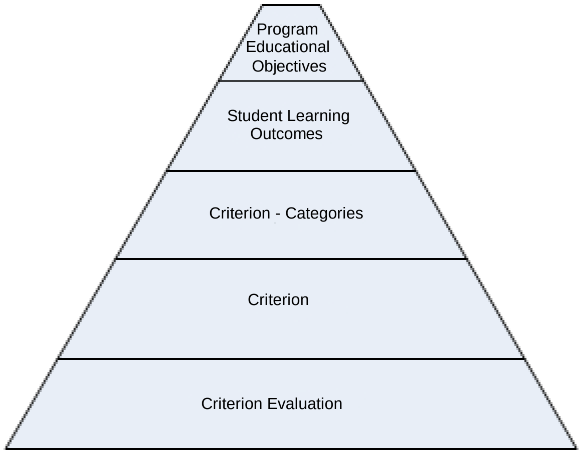


Figure 1. Project overview

3 Development

The public view is available to all user types. Access to the information requires no login. Some of the information that is available includes the university, college, and department mission statements and goals. Mission statements set the tone for the purpose of each entity, whereas goals are used to provide a direction and path of intention for each entity. The objectives section displays the Department, College, and University program educational objectives and program outcomes. The objective-outcome table view lists each program educational objective with a corresponding list of student learning outcomes. A part of this table is shown in Figure 2.

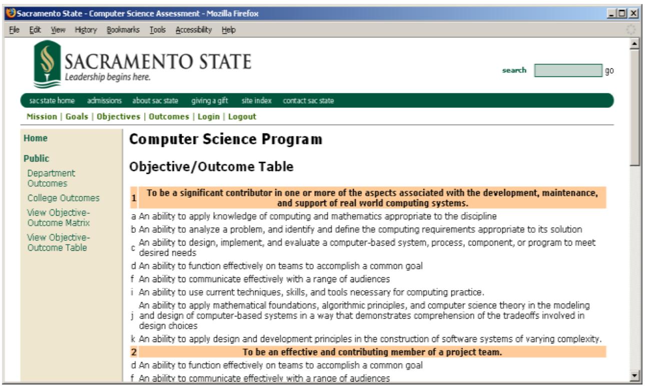


Figure 2. Objective-outcome table

The objective-outcome matrix displays the program educational objectives numbers and the corresponding student learning outcomes letters in a matrix with links to tables describing each objective number and outcome letter, as shown in Figure 3.

.

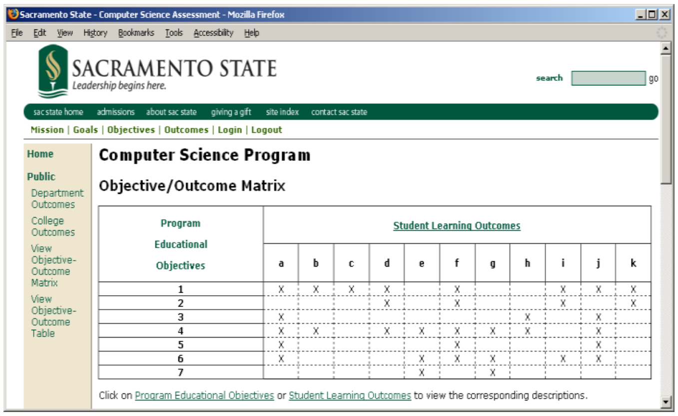


Figure 3. Objective-outcome matrix

The data entry view prepares an evaluation form for printing through the *Prepare Evaluation Form for Print* link. The evaluation form is what evaluators use to score a specific rubric form. The steps include selecting the number of students per group, selecting the course, and the appropriate rubric form. The last step is to select the semester and year in which this evaluation is taking place. After the form is submitted, an evaluation form is generated and displayed.

The faculty view has restricted access and requires a login. This view provides access to the rubric forms and data analysis tool. The faculty view for objectives includes the ability to view attachments. Attachments are uploaded documentation that is associated with one or more objectives. These attachments may be the result of an industrial visit, surveys, or other types of assessment data.

Under the performance criteria heading, a drop-down menu of available rubric forms are listed for faculty viewing. The user selects the desired rubric form and clicks submit. The selected rubric form displays the program and rubric form title followed by sections of categories. Each category lists the criteria that are associated with that category. Each criterion will contain the descriptions of each score value along with whether the criterion is to be scored as a team or individually.

The data analysis tool uses step-by-step instructions to assist the user. Since the data analysis is tied to outcomes, the first step is to select the student learning outcome. The second step is to select a specific course from the drop-down menu. The third step is to select the semester and year. The final step is to select the result type. The result type options are mean, median, mode, percentage, and raw data.

When the result type, mean, is selected, the application displays the total score value, the number of scores entered, and the mean value for each criterion. When the result type, median, is selected, the application displays the median value for each criterion. When the result type, mode, is selected, the application displays the mode value for each criterion.

When the result type, percentage, is selected, the application displays the total number of scores for each criterion followed by the percentage of the score values. The percentage is broken down into the percentage of score values equal to four (% of 4: Exceeds criterion), the percentage of score values equal to three (% of 3: Meets criterion), the percentage of score values equal to two (% of 2: Progressing to criterion), and the percentage of score values equal to one (% of 1: Below expectations). The Department has established a minimum standard of 75% to determine if a criterion is met. That is, the percentage of 3s and 4s must be greater than or equal to 75% of the total assessment scores in order for a criterion to be satisfied.

When the result type, raw data, is selected, the application displays the raw data. This includes the criterion ID and name, instructor name, team ID, date, evaluator type, and score value.

Once a data result type is selected, the student learning outcome, course, semester, and year are displayed in a table at the top of the page followed by the data results. After the data results, the user can select another result type to be displayed. These results will use the same student learning outcome, course, and semester/year selections. The user can also click reset to start over and select a new student learning outcome, course, semester, and year combination.

The data layer of this three-tier web application architecture utilizes a MySQL database. The database is designed in a top-down manner, as shown in Figure 4. At the top is the objective table. A given objective may have one or more attachments associated with it. In addition, an attachment is associated with one or more objectives. Therefore, a mapping table must exist between the two tables. This breaks a many-to-many table relationship into two one-to-many table relationships. The same issue exists between objectives and outcomes. Each objective is associated with at least one outcome, and each outcome is associated with one or more objectives. To compensate for the many-to-many relationship, the objective_outcome_map table maps the relationships between the objective and outcome tables and creates two one-to-many table relationships.

The rubric forms are based on a single outcome which is comprised of multiple categories. Various performance criteria are grouped into categories. This is the relationship between objectives, outcomes, categories, and performance criteria.

Rubric forms are used to evaluate student performance on specific criteria. A single score rates a specific criterion for a specific course, semester, and student and/or team. Each score contains the score value and foreign keys to the criterion, header information, and team, if applicable. The header information is a collection of non-unique elements, such as course, instructor, semester, year, and date of the evaluation. These non-unique elements are grouped in the evaluation_header table to reduce redundancy in the database.

A given course may be taught in a group setting where teams are evaluated in addition to the individual. To accommodate these cases, there is a weak link between the student teams and the course. A team must be associated with a course, but a course may not have any teams; therefore teams are dependent on the course.

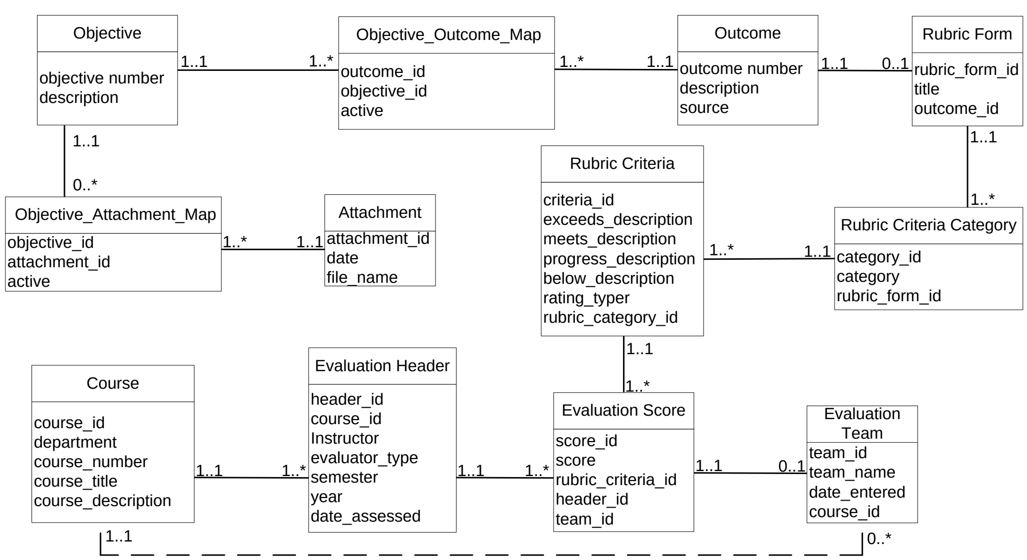


Figure 4: Database design

4 Implementation and Results

This system was implemented using prior assessment data. Top level information on mission statements, program educational objectives, student learning outcomes, and their links have been incorporated. It is anticipated that much of our recent assessment data from 2006 to 2009 will soon be incorporated into this system.

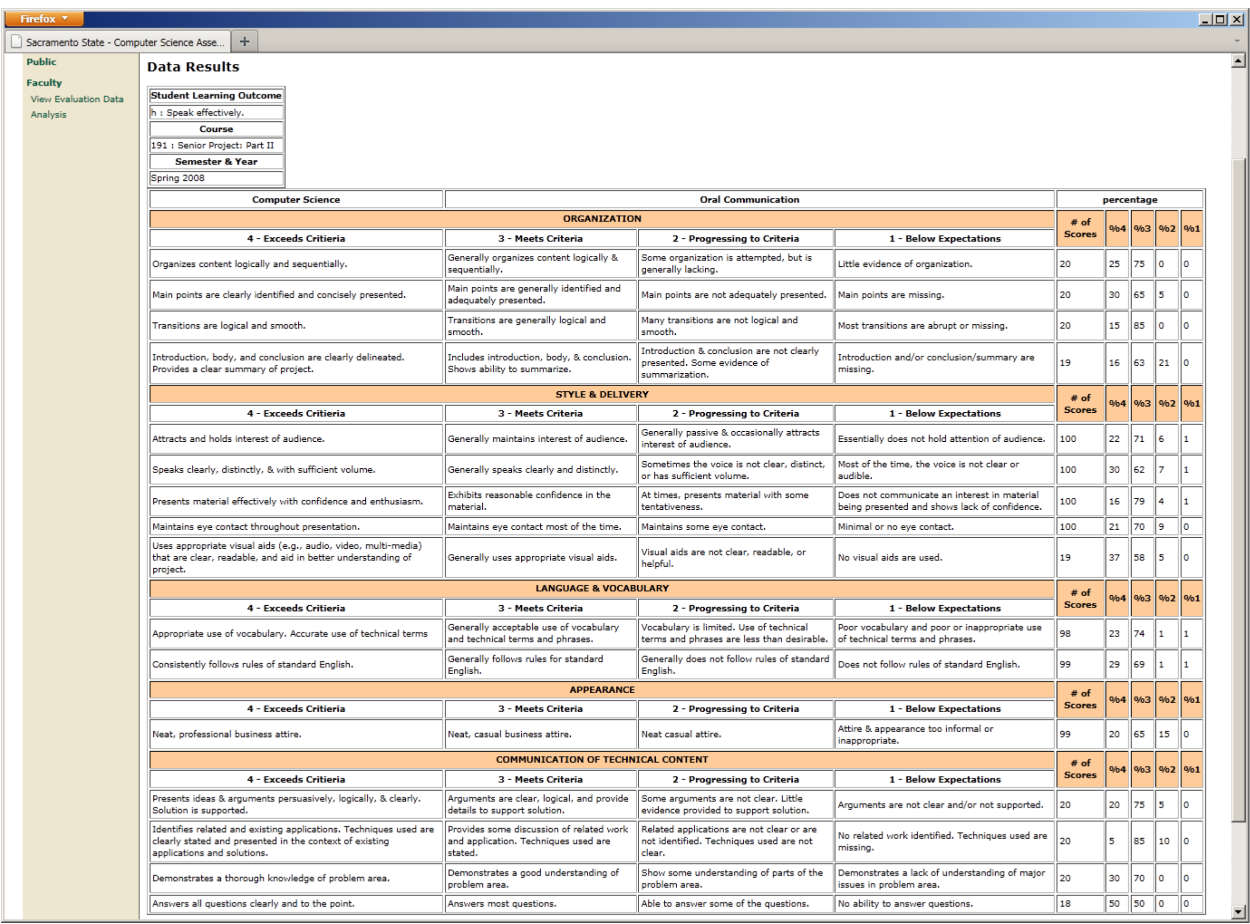
As an example, the results of our Spring 2008 assessment of student learning outcome: Effective Oral Communication are provided in Figure 5. This figure shows the rubric used to evaluate senior project presentations and its results. The rubric assessed student oral presentation skills in terms of the following categories:

Organization
Style and delivery
Language and vocabulary
Appearance
Communication of technical content

Within each of these five categories, between one to five performance criteria were identified. Each performance criterion was described for each of the possible scores of 1, 2, 3, or 4 (1 - below expectations, 2- progressing to criteria, 3 - meets criteria, and 4 - exceeds criteria). In this example, under Organization category, there are four performance criteria. For each criterion, a score of 4 (under Organization) means that a presenter accomplished the following:

Organizes content logically and sequentially
Main points are clearly identified and concisely presented
Transitions are logical and smooth
Introduction, body, and conclusions are clearly delineated; provides a clear summary of project

The second semester senior project teams consisted of 5 or 6 members each. Differences in the number of scores are attributed to whether individual team members or the entire team as a whole were evaluated for that criterion. The last five columns provide the data results. *Number of Scores* specifies the number of evaluators/students' scores. The results of this assessment indicate that our senior-level students demonstrate effective oral presentation skills.



Data Results

Student Learning Outcome
h : Speak effectively.
Course
191 : Senior Project: Part II
Semester & Year
Spring 2008

Computer Science	Oral Communication			percentage				
ORGANIZATION				# of Scores	%4	%3	%2	%1
4 - Exceeds Critieria	3 - Meets Criteria	2 - Progressing to Criteria	1 - Below Expectations					
Organizes content logically and sequentially.	Generally organizes content logically & sequentially.	Some organization is attempted, but is generally lacking.	Little evidence of organization.	20	25	75	0	0
Main points are clearly identified and concisely presented.	Main points are generally identified and adequately presented.	Main points are not adequately presented.	Main points are missing.	20	30	65	5	0
Transitions are logical and smooth.	Transitions are generally logical and smooth.	Many transitions are not logical and smooth.	Most transitions are abrupt or missing.	20	15	85	0	0
Introduction, body, and conclusion are clearly delineated. Provides a clear summary of project.	Includes introduction, body, & conclusion. Shows ability to summarize.	Introduction & conclusion are not clearly presented. Some evidence of summarization.	Introduction and/or conclusion/summary are missing.	19	16	63	21	0
STYLE & DELIVERY				# of Scores	%4	%3	%2	%1
4 - Exceeds Critieria	3 - Meets Criteria	2 - Progressing to Criteria	1 - Below Expectations					
Attracts and holds interest of audience.	Generally maintains interest of audience.	Generally passive & occasionally attracts interest of audience.	Essentially does not hold attention of audience.	100	22	71	6	1
Speaks clearly, distinctly, & with sufficient volume.	Generally speaks clearly and distinctly.	Sometimes the voice is not clear, distinct, or has sufficient volume.	Most of the time, the voice is not clear or audible.	100	30	62	7	1
Presents material effectively with confidence and enthusiasm.	Exhibits reasonable confidence in the material.	At times, presents material with some tentativeness.	Does not communicate an interest in material being presented and shows lack of confidence.	100	16	79	4	1
Maintains eye contact throughout presentation.	Maintains eye contact most of the time.	Maintains some eye contact.	Minimal or no eye contact.	100	21	70	9	0
Uses appropriate visual aids (e.g., audio, video, multi-media) that are clear, readable, and aid in better understanding of project.	Generally uses appropriate visual aids.	Visual aids are not clear, readable, or helpful.	No visual aids are used.	19	37	58	5	0
LANGUAGE & VOCABULARY				# of Scores	%4	%3	%2	%1
4 - Exceeds Critieria	3 - Meets Criteria	2 - Progressing to Criteria	1 - Below Expectations					
Appropriate use of vocabulary. Accurate use of technical terms	Generally acceptable use of vocabulary and technical terms and phrases.	Vocabulary is limited. Use of technical terms and phrases are less than desirable.	Poor vocabulary and poor or inappropriate use of technical terms and phrases.	98	23	74	1	1
Consistently follows rules of standard English.	Generally follows rules for standard English.	Generally does not follow rules of standard English.	Does not follow rules of standard English.	99	29	69	1	1
APPEARANCE				# of Scores	%4	%3	%2	%1
4 - Exceeds Critieria	3 - Meets Criteria	2 - Progressing to Criteria	1 - Below Expectations					
Neat, professional business attire.	Neat, casual business attire.	Neat casual attire.	Attire & appearance too informal or inappropriate.	99	20	65	15	0
COMMUNICATION OF TECHNICAL CONTENT				# of Scores	%4	%3	%2	%1
4 - Exceeds Critieria	3 - Meets Criteria	2 - Progressing to Criteria	1 - Below Expectations					
Presents ideas & arguments persuasively, logically, & clearly. Solution is supported.	Arguments are clear, logical, and provide details to support solution.	Some arguments are not clear. Little evidence provided to support solution.	Arguments are not clear and/or not supported.	20	20	75	5	0
Identifies related and existing applications. Techniques used are clearly stated and presented in the context of existing applications and solutions.	Provides some discussion of related work and application. Techniques used are stated.	Related applications are not clear or are not identified. Techniques used are not clear.	No related work identified. Techniques used are missing.	20	5	85	10	0
Demonstrates a thorough knowledge of problem area.	Demonstrates a good understanding of problem area.	Show some understanding of parts of the problem area.	Demonstrates a lack of understanding of major issues in problem area.	20	30	70	0	0
Answers all questions clearly and to the point.	Answers most questions.	Able to answer some of the questions.	No ability to answer questions.	18	50	50	0	0

Figure 5. Data analysis - percentage result type

In general, the department is pleased with this system. It provides a centralized, public location for the department's assessment efforts that will be available to students, faculty, assessment coordinators, and others. This system provides an assessment tool that is easy to use, maintain, and update. Program objectives and student learning outcomes are clearly stated. Rubrics and results are linked with appropriate outcomes. With this system, faculty members are better informed and, ultimately, will be more engaged in the department's assessment efforts. Future improvements include incorporating additional data analysis tools, providing graphical results, and providing an enhanced menu-driven user interface. Over time, it would be useful to be able to analyze how successful students are in satisfying a particular outcome by observing a history of assessment results for that outcome. This would provide valuable information on continuous improvement. As an initial prototype, this system provides the foundation for future innovations in the area of an integrated environment supporting data analysis, which also links program objectives and student outcomes helping the department organize and manage its assessment efforts.

5 References

[1] Onitsuka, Lynne M. Development of a Data Analysis Tool for Assessment. Master's Project. California State University, Sacramento. May 2009.

[2] ABET Board of Directors. (2007, Nov.). Criteria for Accrediting Engineering Programs. ABET. Inc. Baltimore, MD. [Online]. Available: http://www.abet.org/Linked Documents-UPDATE/Criteria and PP/E001 08-09 EAC Criteria 12-04-07.pdf

Accountability for Quality in Higher Education

P. Hamilton[1], **and N. Passos**[2]
[1]Graduate School, Midwestern State University, Wichita Falls, TX
[2]Department of Computer Science, Midwestern State University, Wichita Falls, TX

Abstract – *The pre-college education system in the United States seems to be ineffective in the areas of mathematics, sciences, and verbal ability. According to comparative tests applied in different nations, American students generally qualify below several other countries. This trend is also noticed at the college level, considering the number of remedial classes being offered in any given semester. In order to improve the quality of learning among college graduates, local, state, and national assessment and accreditation efforts have been implemented. Many of these efforts derive from quality improvement models in manufacturing. However, it seems that such efforts in higher education are not producing quality in proportion to proponents' claims or to their cost. Similar efforts have been applied in health care with equally disappointing results. This study focuses on how much administrators may be relying on easily captured but flawed data for "knowledge" of what is actually happening in higher education.*

Keywords: accreditation, higher education, health care, Deming, institutional ethnography, assessment

1 Introduction

The rapid changes in technology that resulted in the proliferation of personal computers at the end of the twentieth century brought doubts about how America would compete in this new environment. One of the main aspects on the preparation for this new world is the correct adjustment of the learning goals established by educational institutions. Under this rhetoric, the concept of assessing the quality of student education became a central point of discussion. As a response to such changes, new assessment policies have been created, focused on the general idea of quality improvement. This study shows that such concepts may have been wrongly interpreted when typical ideas of quality assessment in manufacturing plants seem to have surfaced in the evaluation of the education process. In particular, this study focuses on the state of Texas, where information was readily available to the authors.

The following excerpt was taken from "Closing the Gaps," a report produced by the Texas Higher Education Coordinating Board [3] in 2001: "*Texas is profiting from a diverse, vibrant and growing economy. Yet this prosperity will always depend on steps taken quickly to ensure an educated population and workforce for the future. Unfortunately, at present, the proportion of Texans enrolled in higher education is declining. Too few higher education programs are noted for excellence and too few higher education research efforts have reached their full potential.*" The same publication outlined the steps to be taken to improve the level of knowledge of the Texas population, by increasing the number of college graduates as well as investment in research. However, the most important goal of such a report is to achieve excellence. As a consequence of such a goal, some sort of measurement is usually adopted to allow a comparison between institutions. One such measure is verbal ability.

Nie and Golde provide evidence suggesting that verbal ability, rather than being improved by education, is an attribute that enables students to succeed in schools and universities [8]. Thus, while education and verbal ability have a strong positive relationship, the direction of causality may be counterintuitive. Nie and his colleagues at the Stanford Institute for the Quantitative Study of Society have illustrated how recent trends in increased education in the U.S. population have not, for example, resulted in increased verbal ability of the population. As a matter of fact, in recent years, as a greater proportion of the U.S. population entered higher education, the average verbal ability scores on college entrance examinations have remained relatively unchanged [15]. This is made even more troubling by the cost of higher education. The average cost of an undergraduate education in the U.S. is $16,000 per year for tuition, books, room and board at public universities and $37,000 per year at private universities [15]. The public is questioning the quality of higher education.

The health care system in the United States is also in a crisis of quality. Surveys reported in 2002 showed the system not to be as safe as it should be even though the U.S. devotes 16% of its gross domestic product (gdp) to health care [16]. Those reports show that at least 44,000 people, and perhaps as many as 98,000 people, were estimated to die in U.S. hospitals each year as a result of medical errors that could have been prevented [9]. Even using the lower estimate, preventable medical errors in hospitals exceeded deaths attributed to such feared threats as motor-vehicle wrecks, breast cancer, and AIDS. Since such results were assumed related to the quality of services provided, then that quality has been challenged. Quality is partially defined by the U.S. Agency for Healthcare Research and Quality as doing the right thing, at the right time, in the right way, for the right person, and having the best possible results [5]. Borrowing this simple definition, one might state that educational quality would be: learning the right things, at the right times, in the right ways, from the right persons, and having the best possible results. However, when it comes to measuring either quality in either education or health care, great many problems arise. This study focuses on the evaluation of quality measurements in place today and their apparent effectiveness. The next section describes the concepts related to quality improvement, usually developed in a business administration environment. The paper proceeds with an examination of quality improvement practices and results ending with a summary and suggestions for different approaches to quality assessment.

2 Background

Quality Improvement (QI) is a term first coined when corporations began looking at ways to streamline and improve processes and systems. Other similar terms include Total Quality Management (TQM), Quality Control, Continuous Quality Improvement (CQI), and Six Sigma. W. Edward Deming contributed significantly to the theory and practice of quality improvement [4]. He is probably best known for his contributions to the auto industry in Japan, following World War II. In 1950, Deming was invited to speak to the Union of Japanese Scientists and Engineers and numerous other groups such as the Japan Medical Association on the statistical control of quality. His ideas were implemented enthusiastically and his contributions to understanding and managing quality are still honored worldwide by the prestigious Deming Prize.

Here in the United States, traces of Deming's PDCA model, shown on Figure 1, can be found in virtually every approach to quality assurance and improvement. Deming changed "check" to "study" in later versions of his model in order to stress the scientific approach to quality management.

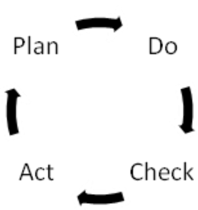


Figure 1. Deming's PDCA model

Approximately 30 years after his return from Japan, Deming wrote a book, *Out of Crisis*, aimed at managers [4]. This later work stressed 14 points for managers to adopt in order to maintain market share, preserve jobs, protect investments, and ensure future dividends. Those 14 points are listed in Table 1. It is noticeable that there were no hints on how to measure quality. However, in the industry, everyone was well informed that they would be accountable for their results.

The word accountability became commonly used in the English language in the late 18th century. One definition of accountability is "the quality or state of being accountable; *especially*: an obligation or willingness to accept responsibility or to account for one's actions" [1]. It is clear that the coincidence of the use of the word accountability and the industrial revolution implied the idea of making sure that all manufactured products would have the same characteristics, such as shape, weight, components, connections, smoothness, etc.

Table 1. Deming's 14 Points for Managers

1. Create constancy of purpose.
2. Adopt a new philosophy.
3. Cease inspection. Require evidence.
4. Improve the quality of supplies.
5. Continuously improve production.
6. Train and educate all employees.
7. Supervisors must help people.
8. Drive out fear.
9. Eliminate boundaries.
10. Eliminate the use of slogans.
11. Eliminate numerical standards
12. Let people be proud of their work.
13. Encourage self-improvement.
14. Commit to ever-improving quality.

Peter Miller from the London School of Economics and Political Science, suggests accounting has become a body of expertise preeminently concerned with the exacting of responsibility from individuals rendered calculable and comparable [7]. He states "Increasingly, management accounting…is a practice deployed not only in the private sector, manufacturing industry, not only in the privately owned and managed service industries, but also in areas of the public sector as diverse as health, education, and social services." Miller goes on to describe ways in which accounting practices "…make it possible to render visible both the near and the distant activities of individuals, to calculate the extent to which they depart from the norm or performance, and to accumulate such calculations in computers and files and compare them."

3 Quality improvement

Individual and institutional accountability require assessment. Current efforts to assure quality in higher education are centered in assessment and accreditation processes. In particular, Texas higher education is under the auspices of the Texas Higher Education Coordinating board as well as the Southern Association of Colleges and Schools (SACS).

The Accreditation Criteria of the Southern Association of Colleges and Schools includes the following evaluation aspects [10]:

- What are the institution's processes for systematic, ongoing, integrated, research based reviews that result in continuous improvement?

- How does the institution demonstrate a sustained, documented history of planning evaluation cycles, including the use of results for improvement, to accomplish the institution's mission?

- What is the evidence that data from various sources concerning the effectiveness of programs and services are being used to make decisions for improvement?

Such criteria immediately imply self assessment being conducted by each institution. However, any approach to quality evaluation based on checking or studying processes and outcomes rests on the availability, accessibility, and acceptability of data. According to the humorous, but no less accurate, *Finagle's Law of Data* [6]

- The information you have is not the information you want.

- The information you want is not the information you need.

- The information you need is not the information you can obtain.

In order to illustrate Finagle's Law, consider a "real life" example: one of the authors went to a restaurant to have breakfast. The menu board above the cash register listed scrambled eggs. However, when ordering scrambled egg white (there was a wish of limiting cholesterol intake), the server replied he could not provide that option. The order was then changed to two scrambled eggs. The purchase was completed and a numbered flag taken to the table while the server keyed the order. The purchase was recorded on a point of sale (POS) cash register/computer. The website for the specific POS states: *"Today business owners face more pressure from tough competition, new business concepts, and eroding margins than ever before. How can you continuously deliver rapid, dependable service during peak hours, manage demands for greater product selections, and at the same time control escalating costs?"* [11]

This type of data collection, communication, and quality monitoring system, allows information about items ordered to be transmitted instantly to the kitchen where preparation begins and the time until the order is delivered to the customer is tracked. The restaurant owner later extracts data from the system in order to analyze food cost; identify trends in items sold, and monitor time from point of sale to delivery of food to the diner's table. Important information about quality of the food preparation, food cost, and employee performance are tracked by information obtained from the computer. Anyway, the food arrived rapidly but the tray contained two very large platters of scrambled eggs. Talking to the worker who delivered the food, the author discovered that the person taking the order simply pressed the scrambled egg key twice. For each order, the cook dipped out two ladles-full of egg mixture into a skillet. Each ladle held the equivalent of two eggs. Therefore, the customer's original desire for two scrambled egg whites had been modified and "worked up" in a system designed for efficiency, quality assessment and process control. The actual outcome was an 8-egg breakfast order that wasted employees' time to explain the mistake and process a refund; increased food cost with the wasting of six uneaten eggs, and diminished validity of information needed for staving off eroding margins.

One might say operator error was at the root of this mismatch of customer desire to actual outcome. It is obvious that better training in order processing might have prevented the negative outcomes. However, it is also clear that there are ways in which the very techniques used to obtain and analyze quality data can create erroneous data and cloud the "distant" understanding of what actually occurs, referred to by Miller. That should be especially true when knowledge about quality of higher education is the goal.

Schray, in a U.S. education commission report on the future of higher education, wrote that one method of evaluating educational results would be performance outcome measures [12]. Schray stated that "The strongest emphasis would be placed on the demonstration by institutions and programs that they are producing results, especially evidence of student learning. The framework would report student learning based on standards for valid and reliable assessment. The framework would also contain a set of comparable performance measures that include student learning that would be tailored according to institutional mission and program so they can be used for both accreditation and public reporting and consumer profiles." The language used in the commission report sounds very much like the numerical standards and reliance on inspection that Deming discouraged and the type of "accountability" described by Miller in which persons and institutions become comparable and, thus, calculable.

Schray's recommendations are very similar to those already in place to evaluate health care in the U.S. Many states require public reporting of cost, outcome, and comparisons of types of care delivered. Particularly, in Texas, hospitals are required to submit comprehensive information on all discharged inpatients attended or treated by physicians. Discharge data reports follow the national uniform billing data element specifications and additional state-required data elements. After data files are received by Texas health care information collection (THCIC), they are reviewed to determine if they conform to data specifications.

The scale of the reporting is massive, the file examined in this study included information on 1,814,435 outpatient surgical and radiological procedures performed in Texas hospitals and ambulatory surgery centers during October, November and December 2009, the data elements appear to be specific and standardized across hospitals. Hospital data are submitted quarterly and hospitals are permitted to include explanations and disclaimers related to their reports. The report under exam included 99 pages of disclaimers and explanations such as the following:

- *…Conclusions drawn from the data are subject to errors caused by the inability of the hospital to communicate complete data due to reporting form constraints, subjectivity in the assignment of codes, system mapping, and normal clerical error. The data are submitted by hospitals as their best effort to meet statutory requirements.*

- *….Because of system constraints, all admissions on the encounter records are reported as urgent. The data report also includes emergency admissions.*

- *…Patient diagnoses and procedures for a particular hospital stay are coded by the hospital using a universal standard called the International Classification of Disease, or ICD 9 CM. This is mandated by the federal government. The hospital complies with the guidelines for assigning these diagnosis codes; however, this is often driven by physician's subjective criteria for defining a diagnosis.*

Similar to the person searching for the right key to press to accommodate an order for two scrambled eggs, the hospitals struggle to make the work they actually perform calculable and comparable.

On the educational side, Derek Bok, Harvard University president emeritus, reports that he introduced the Collegiate Learning Assessment test at Harvard in order to obtain student evaluations and he admits that even such a test has very significant problems in terms of its validity [2].

Based on what can be learned from disappointing quality reporting in health care, from disappointing assessment in education, and from personal experience, one may conclude that mandated reporting and accounting management cannot provide information on quality that is suitably available, accessible, and acceptable. The employment of ethnographic techniques is necessary to examine a more comprehensive scope of quality in higher education.

One such type of ethnography is Institutional Ethnography developed by Canadian Dorothy Smith [13]. Smith's approach makes possible a thorough understanding of the actual happenings in social settings rather than the faint traces and sometimes misleading distant knowledge of happenings. Institutional Ethnography does this by focusing on how the institution "works up" information, how rules created elsewhere (in order to control work) actually operate, whose standpoint is valued in creating knowledge about the institution and its output, unintended consequences of institutional rules, how individuals do/do not "work around" institutional rules, and how the information gathered by the institution determines the rules, consequences and work of individuals.

Ethnographic approaches are particularly useful for deriving knowledge of the contexts in which education takes place. For example, recent studies show that undergraduate students are consuming more than 50 % of their time in socializing and only 16 % in studying [14]. It would be important to understand how those findings impact on learning.

The new social settings have changed the college student habits, resulting in a very low dedication to study. Therefore, student learning assessment must be constantly re-evaluated due to the fast change in technology, where computerized social networks are replacing physical proximity and face to face contact.

4 Summary

Any educational setting, hospital, factory, or fast food restaurant operates on the basis of institutional knowledge. The more complex the organization, the more its accountability is tempted to rely solely on easily captured data which contains flaws derived from attempts to make the data standardized and, thus, comparable. Consequently, flawed institutional knowledge is used from a distance to coordinate people, other resources, and activities, making them manageable and accountable across space and time but not necessarily of higher quality.

In summary, the credibility of educators is decreasing because citizens feel costs of higher education are high and do not result in commensurate improvement in learning. Educators engage in complex and constantly changing work in a "market-driven" society, and now they are being required to be more "accountable", to take up the task of producing what is desired in the marketplace and what can be made standardized and comparable. Institutional Ethnography is one means to go beyond "accountability" for Quality Improvement and to examine the meta-structures that affect students and faculty, not just what they "produce" but what those outcomes could be.

Over thirty years ago, Deming proposed that quality comes from total commitment from the manager to the workers, who personally take pleasure and pride in doing their best. In higher education that would mean state legislatures, boards of education, university regents, presidents, administrators, faculty and students all sharing responsibility for learning. To achieve quality all must focus not simply on becoming comparable, but rather on the pleasure and pride which Deming demonstrated is the foundation on which quality rests.

5 References

[1] Accountability (2011) Merriam-Webster Dictionary. [Online]. Available: http://www.merriam-webster.com/dictionary/accountability

[2] D. Bok, "All I Really Need to Know, I didn't learn in college" in *Miller-McCune*, March-April 2011, pp. 44-46.

[3] *Closing the Gaps: The Texas Higher Education Plan*, (2001) Texas Higher Education Coordinating Board [Online]. Available: http://www.thecb.state.tx.us/reports/

[4] W. E. Deming, *Out of Crisis*. Cambridge, MA: Institute of Technology Center for Advanced Engineering Study, 1986.

[5] J. M. Eisenberg, (1997, October 28) *Testimony on Health Care Quality Before the House Subcommittee on Health and the Environment*, [Online]. Available: http://www.ahrq.gov/news/test1028.htm

[6] *Finagle's Law* (2011) WorldLingo Translations LLC [Online]. Available: http://www.worldlingo.com/ma/dewiki/en/Finagles_Gesetz

[7] P. Miller, "Accounting and Objectivity: The Invention of Calculating Selves and Calculable Spaces," in *Rethinking Objectivity*, A. Megill (ed.), Durham, US: Duke University Press, 1994, pp. 239-264.

[8] N. Nie, and S. Golde, "Does education really make you smarter?" in *Miller-McCune*, June-July 2008, pp. 56-64.

[9] *Patient Safety: Current Statistics*. (2011) Arrowsight, Inc. [Online]. Available: http://www.patientsafety focus.com/patient-safety-current-st.html.

[10] *The principles of accreditation: Foundations for quality enhancement. (*2008) Southern Association of Colleges and Schools, [Online]. Available: http://www.sacscoc.org/pdf/2008PrinciplesofAccreditation.pdf.

[11] *QTouch 2. Product Catalog* (2011) Quoriun. [Online]. Available: http://www.quorion.de/Cash Registers/Q-Touch-pos-system.htm

[12] V. Schray. (2006) "Assuring quality in higher education: Recommendations for improving accreditation," US Department of Education. [Online]. Available: http://www.ed.gov/about/bdscomm/list/hiedfuture/reports/schray2.pdf.

[13] D. E. Smith, *Institutional ethnography: A sociology for people*. Toronto: Alta Mira Press, 2005.

[14] *Texas Health Care Information Collection: Center for Health Statistics: Public Use Data File* (2011), [Online]. Available: http://www.dshs.state.tx.us/thcic/default.shtm

[15] "What happened to Academic Rigor" in *Miller-McCune*, March-April 2011, pp. 44-46.

[16] *OECD Health Data 2010*. Organization for Economic Co-operation and Development *[*Online]. Available: http://www.oecd.org/dataoecd/46/2/38980580.pdf

Realigning for Retention: Investigating the Interests, Attitudes, and Academic Preparedness of College Engineering Students[1]

Deborah L. Worley[1], Naima Kaabouch[2], Jeremiah Neubert[3], and Mohammad Khavanin[4]
[1]Department of Educational Leadership, University of North Dakota, Grand Forks, ND, USA
[2]Department of Electrical Engineering, University of North Dakota, Grand Forks, ND, USA
[3]Department of Mechanical Engineering, University of North Dakota, Grand Forks, ND, USA
[4]Department of Mathematics, University of North Dakota, Grand Forks, ND, USA

Abstract - *This paper describes the results of a baseline survey that was administered to undergraduate students enrolled in introductory engineering courses at a mid-size research institution located in the Midwest. The purpose of the survey was to establish a general understanding of engineering-related interests, attitudes, and academic preparedness from the college student perspective. The data were used to inform our development of a calculus course designed to improve retention rates in engineering as part of a National Science Foundation (NSF) Course Curriculum and Laboratory Improvement (CCLI) grant.*

Keywords: engineering attitudes, academic preparedness, retention

1 Introduction

There are a large number of engineers moving towards retirement in the present day, and the demand for new engineers in the United States will soon exceed supply [1]. This projection becomes increasingly alarming when coupled with reports of the nation's weakening capacity in core competencies required for success in engineering study and subsequent careers [1][2]. At the heart of the concern is dwindling student knowledge levels, skills, and abilities in key foundational math and science courses. An analysis that revisits the "gathering storm" reveals that little progress has been made in recent years (2005-2010) to avert this impending crisis [2]. Indeed, the projected shortage of human capital, in the form of qualified, well-trained engineers, is coming to fruition.

To avert a shortage of skilled engineers it is imperative that every incoming engineering student has the best possible opportunity to obtain a degree. An examination of attrition rates at engineering schools around the country reveal that three out of every ten students entering engineering programs leave the major in the first year [3][4][5][6]. The attrition rates are even more troublesome for incoming minority and female students in engineering, where numbers of students in these populations leaving the academic field of engineering can exceed six out of ten [6][7].

To quote the National Academy of Science report, "Rising Above the Gathering Storm: Energizing and Employing America For A Brighter Economic Future":

> The undergraduate years have a profound influence on career direction, and they can provide a springboard for students who choose to major and then pursue graduate work in science, mathematics, and engineering. However, many more undergraduates express an interest in science, mathematics, and engineering than eventually complete bachelor's degrees in those fields. A focused and sizeable national effort to stimulate undergraduate interest and commitment to these majors will increase the proportion of 24-year-olds achieving first degrees in the relevant disciplines. [1].

Projections of persistence in STEM fields such as engineering are not favorable for students at any level of study [8][9], which notably translates to low rates of transition to working in the field of engineering upon degree completion [10][11]. This trend is unfortunate, considering that recent U.S. Bureau of Labor Statistics reports indicate that occupations with optimistic projections for employment growth are centered on technical fields such as engineering [12]. In response to unacceptably high attrition rates in engineering combined with the decline in qualified engineers for the workforce, institutions of higher learning must capitalize on opportunities to educate the next generation of engineers. Faculty, administrators, and staff must commit to preparing, providing, and presenting students with optimal chances to succeed in engineering. In order to do so, it is imperative to know and understand the experiences of students who express interest in studying engineering, paying special attention to curricular components that are known to be discouragers to pursuing engineering as a major, such as calculus [4].

[1] Funded by the National Science Foundation (NSF) Course Curriculum Laboratory Improvement Grant (Project DUE-0942270)

This paper describes the results of a baseline survey that was administered to undergraduate students enrolled in introductory engineering courses at a mid-size research institution located in the Midwest. The purpose of the survey was to establish a general understanding of engineering-related interests, attitudes, and academic preparedness from the college student perspective. We sought the answers to questions concerning student motivations for pursuing engineering as an academic major as well as a career path. We also asked questions about confidence and comfort levels of students in a variety of academic disciplines in an effort to provide a baseline for comparison. Finally, we queried current college-level engineering students about their academic preparation in the area of mathematics, a requisite field of study which all successful engineers must have mastery.

2 Methodology

The Interest, Attitudes, and Preparedness (IAP) Survey for the Study of Engineering was designed to gather information about students' interests, attitudes, and academic preparedness related to studying engineering at the college level. There were three sections to the survey. Section one contained 21 questions that concerned students' interests and attitudes related to engineering. In Section two, there were five questions about students' study of mathematics in high school as well as at the beginning of their college careers. Section three was dedicated to collecting demographic statistics about the students; there were four questions in this last section.

The IAP survey instrument was reviewed and approved by the Institutional Review Board (IRB) at the University where it was administered. Upon certification that the study met the human research subject protections and obligations as required by law and University policy, members of the research team responsible for collecting data visited three different courses to administer the survey: *Introduction to Mechanical Engineering*, *Graphical Communications*, and *Introduction to Engineering*. These courses serve entering students who declared Civil, Electrical, and Mechanical Engineering as their academic majors, as well as students who were interested in engineering, but had not yet declared a discipline. The *Introduction to Mechanical Engineering* and *Graphical Communications* courses provide students with an engineering design experience and basic drafting skills. The *Introduction to Engineering* course is a required course for any student interested in pursuing an engineering major; it is designed to provide an overview of the engineering-specific discipline (i.e., electrical engineering or mechanical engineering).

3 Results

The survey was administered to undergraduate students in three courses during the Fall 2010 semester (n=164). Students' participation in the study was voluntary. Those who chose to participate overwhelmingly identified as male (89.7%) and White (93.3%). Only 5.5% of participants identified as female. Other racial or ethnic groups represented in the sample included Hispanic or Latino/a, Asian, and Black or African-American. There was some variation in age of participants, although most were of a "traditional" college age: 55.2% indicated their age as 17-18; 32.1% listed their age as 19-20; 1.2% were 21-22 years of age; and 7.9% indicated that they were 23 years of age or older.

The results of the content-based questions on the IAP survey are organized into two sections: 1) interests and attitudes and 2) academic preparedness. The first section concerning interests and attitudes includes participant self-reporting of: factors related to considering engineering as an undergraduate major; confidence levels in skills and abilities for specific academic subjects; comfort levels in knowing how academic subject area are used in engineering; and overall confidence ranking of abilities to succeed in engineering. The second section, academic preparedness, asked students to report information about their study of mathematics in high school as well as at the beginning of their college careers.

3.1 Interests and attitudes

Students were asked to indicate the extent to which a series of statements was important in their decision to consider engineering as an undergraduate major. Response selection ranged from 1 ("not at all important") to 5 ("Extremely important"). Statements that rated the highest among participants as being important in the decision to consider engineering as an undergraduate major included a good job outlook (mean=4.15, SD=0.87), the challenge of solving engineering problems (mean=3.98, SD=0.74), being good at math (mean=3.84, SD=0.91), being good at science (mean=3.72, SD=0.88), and the opportunity to use engineering to solve societal problems (mean=3.55, SD=1.13). Participants indicated that receiving a message from teachers or advisors at the high school level concerning majoring in engineering at the college level was least important to them when selecting a major. Table 1 summarizes students' responses to each item in this section.

Students were also asked to indicate the extent to which they felt confident in their skills and abilities related to certain academic subjects that are considered to be foundational to the study of engineering: Chemistry, Physics, Calculus, Computers/Computer Science. Although all reported levels of this measure of self-efficacy concerning confidence in abilities were low (less than 41%), respondents expressed the least amount of confidence in their skills and abilities related to calculus (see Table 2).

In a related series of questions, students indicated the extent of their confidence in writing and public speaking/communication skills and abilities (see Table 3). Although not specific to the study of engineering, it is important to recognize the role that writing and public speaking play in the success of individuals as engineers throughout their careers. The results of the IAP survey indicate that students' self-reported confidence levels in writing and public speaking /communication skills and abilities plummeted even further when compared to confidence levels in math and science-related academic subjects.

Table 1
Items of Importance in Students' Decision to Consider Engineering as an Undergraduate Major

STATEMENT	MEAN	STDEV
The job outlook for engineering majors is good.	4.15	0.87
The challenge of solving engineering problems appeals to me.	3.98	0.74
I am good at math.	3.84	0.91
I am good at science.	3.72	0.88
Engineering provides opportunities to solve societal problems.	3.55	0.97
A family member or friend is an engineer.	2.81	1.48
Majoring in engineering will prepare me for graduate school or medical school.	2.65	1.29
Engineering is the most challenging undergraduate major.	2.56	1.13
A high school teacher or advisor told me to major in engineering.	2.02	1.10

Table 2
Extent of Confidence in Skills and Abilities in Math and Science-Related Academic Subjects

	Chemistry %	Physics %	Calculus %	Comp Sci %
Not at all confident	3.0	3.0	6.1	3.6
Somewhat confident	17.6	17.0	20.6	14.5
Confident	40.6	37.6	27.9	35.8
Very confident	30.9	28.5	29.7	36.4
Extremely confident	7.3	13.3	15.2	8.5

Table 3
Extent of Confidence in Skills and Abilities in Writing and Public Speaking/Communication

	Writing	Speaking
Not at all confident	4.2	6.7
Somewhat confident	26.1	20.0
Confident	33.9	37.0
Very confident	26.7	24.8
Extremely confident	7.9	10.9

The final set of questions in this section on interests and attitudes concerned the extent to which students felt comfortable in knowing how the same math and science-related academic subject areas (Chemistry, Physics, Calculus, Computers/Computer Science) as well as writing, and public speaking/communication are used in engineering (see Tables 4 and 5).

As with confidence levels, comfort levels representing application to engineering were low (less than 45%). Participants indicated that they were most comfortable in knowing how Calculus and Computers/Computer Science are used in engineering. When students were asked to rate their overall confidence in their abilities to succeed in engineering, the majority answered that they felt confident, very confident, or extremely confident. See Table 6 for specific response percentages at all confidence levels.

Table 4
Comfort Levels in Knowing How Math and Science-Related Academic Subject Areas are Used in Engineering

	Chemistry %	Physics %	Calculus %	Comp Sci %
Not at all comfortable	3.6	1.8	3.6	1.8
Somewhat comfortable	21.2	13.3	10.3	6.7
Comfortable	38.8	26.7	23.0	23.0
Very comfortable	29.7	32.7	40.6	44.8
Extremely comfortable	6.1	24.8	21.2	23.0

Table 5
Comfort Levels in Knowing How Writing and Public Speaking/Communication are Used in Engineering

	Writing	Speaking
Not at all comfortable	3.0	4.8
Somewhat comfortable	18.8	16.4
Comfortable	44.2	35.8
Very comfortable	25.5	26.1
Extremely comfortable	7.9	16.4

Table 6
Extent of Confidence (Overall) in Ability to Succeed in Engineering

Confidence Levels	%
Not at all confident	0.6%
Somewhat confident	3.6%
Confident	26.7%
Very confident	44.8%
Extremely confident	23.6%

3.2 Academic preparedness

In terms of academic preparedness needed to study engineering, the IAP survey included five items related to students' study of mathematics prior to enrolling in an introductory engineering course. A majority (53.3%) of respondents took at least 4 math classes in high school, while 36.4% expressed that they took more than four (see Table 7). Also of interest is the performance of the students in terms of their self-reported letter grade averages in their high school math classes. A full 80% of respondents indicated above average academic performance, in that they received "mostly A's" or "A's and B's". Table 8 provides a full percentage breakdown of average grades (self-reported) in high school math classes.

Table 7
Number of Math Classes Taken in High School

Number of Classes	%
1	0.61
2	0.61
3	8.48
4	53.33
more than 4	36.36

Table 8
Average Grade (Self-Reported) in High School Math Classes

Average Grade	%
Mostly A's	37.58
A's and B's	42.42
Mostly B's	13.33
B's and C's	4.85
Mostly C's and below	1.21

In addition to learning about high school math experiences, we were also interested in knowing how many math classes the participants in this study had taken prior to enrolling in an introductory engineering course. Some students indicated either no college-level math class enrollment (18.79%) or completion of one college-level math class (33.33%). The breadth of response frequencies to this question are provided in Table 9. Participants also indicated on the IAP survey the college math course in which they were currently enrolled, thereby delineating co-enrollment with their introductory engineering course. Although precalculus was the most frequent response (36.97%), it is noteworthy that 33.94% of respondents were taking calculus and 11.52% of respondents were taking calculus II in the same semester as their introductory engineering course (see Table 10).

Table 9
Number of College-Level Math Classes Taken

Number of Classes	%
0	18.79
1	33.33
2	23.64
3	13.94
more than 3	9.70

Table 10
Current Math Course

Course	%
None	7.88
Precalculus	36.97
Calculus	33.94
Calculus II	11.52
Other	9.09

4 Discussion

There is some variation in reasons for being interested in engineering, but it is clear from the results of the IAP survey that students taking an introductory engineering course are thinking of engineering as a career focusing, for example, on potential job outlook and being able to use engineering to solve societal problems. This attitude, as expressed by students, presents a unique opportunity for institutions to fuel that interest by helping students make connections to real-world engineering applications throughout their coursework as they study engineering. Engineers are problem solvers ; unless engineering students realize how calculus can be used to solve problems, they have little incentive to invest the time required in this foundational course for meaningful learning to take place.

4.1 Intersecting calculus with engineering

Although slightly more than forty percent of survey participants indicated that they were "very comfortable" in knowing how calculus is used in engineering, there is still room for improvement. Researchers have identified the need to include engineering content and equations in calculus [4][13][14]. By using real world engineering problems as the vehicle to deliver the content student interest can be increased. The benefits of combining engineering concepts and equations with calculus have been demonstrated repeatedly for some time, yet few universities have implemented such a course. The obstacle to creating such a course it that is not apparent where the teaching responsibility lies, mathematics or engineering.

4.2 Engineering student self-efficacy

The IAP survey results revealed information and provided several measures of self-efficacy for students enrolled in introductory engineering courses. In terms of attitudes, survey participants responded positively to the statement, "I am good at math". This affirmation of ability was echoed in the students' high (A's and B's) self-reported letter grade averages from high school. When asked specifically about calculus, however, students reported low levels of confidence in their skills and abilities; confidence levels that were, in fact, the lowest of all academic subjects of which the IAP survey inquired. This lack of confidence is concerning particularly since a significant percentage of students were taking calculus (33.94%) or calculus II (11.52%) in the same semester as their introductory engineering courses.

Another point of concern arises when comparing the students' self-reported confidence levels in calculus with those related to their overall ability to succeed in engineering. As established previously, students felt that they were good at math, but felt less than confident about their abilities in calculus. However, a full 95.1% of respondents to the IAP survey felt confident, very confident, or extremely confident in engineering as a discipline. Perhaps this finding foreshadows inflated confidence levels in engineering abilities or a lack of understanding of how important calculus is to the study of engineering. More research of a predictive nature is needed to fully understand the disparate nature of these self-efficacy measures.

4.3 Making use of results

The Building Engineering and Science Talent (BEST) federal government committee recommendations for undergraduate education in the science, technology, engineering, and mathematics (STEM) fields emphasize, among other factors, personal attention and peer support as integral to helping student succeed in college-level study [1]. Our approach to understanding current college students' interest, attitudes, and preparedness concerning engineering as a discipline as well as a career path provides baseline data for crafting innovative approaches that establish these important foundational building blocks. To that end, we provide an example of how the IAP survey data are used in a developing project.

The current project focus is on completing inaugural steps of a program that minimizes the mathematical deficiencies of students entering calculus and illustrates the importance of calculus as an engineering tool. We recognize the need to connect students, on a personal level, with their academic studies, and to provide a bridge of understanding between calculus and engineering. We also acknowledge an essential component of engineering student success to be connecting with peers. To that end, we embed a peer support component in the project. This project, funded by a National Science Foundation (NSF) Course Curriculum and Laboratory Improvement (CCLI) grant, centers on the design and delivery of a calculus course with an overall goal to improve retention rates in engineering.

4.3.1 Modules

In the context of calculus, creating a section specifically for engineers augmented with engineering-specific content has been shown to benefit students [13][14]. The results of the IAP survey were used to inform our development of a calculus course designed with increasing engineering retention rates in mind by incorporating not only engineering concepts but real world engineering problems, in the form of modules.

The problem sets developed for each module are based on research or consulting projects encountered by professors in the School of Engineering. The modules are constructed around topics that span various engineering disciplines, such as Chemical, Civil, Electrical, and Mechanical Engineering. It is important to have a broad range of topics for two reasons. First, freshman entering engineering may be undecided on a specific area of focus. By providing incoming freshmen with an experience in each area of engineering they will be better able to identify the one that most interests them. Second, in all but the largest schools, offering a section of calculus for each specialization would not be feasible or cost effective. All problems will contain all the engineering background needed to solve the presented problems. Each problem illustrates that calculus is an important tool used by real engineers throughout their careers.

4.3.2 Mentors

We also used the measures of interests, attitudes, and preparedness to help us construct a mentor component to the engineering-specific calculus course. This purposeful course creation provides a built-in support system for students in that they will work in small groups, led by a peer mentor, to work through the real-world engineering problems as they learn calculus.

There are many other benefits to using mentors. By using a diverse group of mentors, for example, the feeling of isolation among underrepresented groups can be eliminated [15]. The benefits of engineering mentors also includes greater retention of minorities and improved academic performance [15][4]. The mentors are in the same peer group as the mentees so they can provide frank and honest advice to the current freshmen about how to succeed. The mentor's guidance on efficient study habits and time management will be extremely valuable as incoming students often lack the required study skills to be successful in calculus [4]. It is expected that engagement in activities such as mentor-led discussions will increase confidence levels as well as knowledge of calculus concepts for enrolled students, thereby raising engineering retention rates in the long term.

5 Conclusion

Many of the students that switch away from engineering cite math/calculus as one of the most influential factors in their decision [4]. Talented minds are being wasted because of a negative attitude and fear of math, specifically calculus. Those students who switch to another major often have deficiencies in their mathematical background or study skills and are not prepared for the rigors of college calculus. To be sure, there is a clear need to support engineering students in new and innovative ways, specifically by combining the use of modules and mentors, in their study of calculus. Students are indeed interested in studying engineering and fulfilling professional roles as engineers upon graduation. By understanding more about students' interests, attitudes, and preparedness about studying engineering in this descriptive study, we are able to construct an approach to studying calculus, a subject area where students feel least confident, that supports retention efforts in engineering.

6 References

[1] National Academy of Science. Rising Above the Gathering Storm: Energizing and employing America for a brighter economic future. Washington, DC: National Academies Press, 2005.

[2] National Academy of Science. Rising Above the Gathering Storm, Revisited: Rapidly Approaching Category 5. Washington, DC: National Academies Press, 2010.

[3] M. Cavalli, L. Stanlake, S. Tolbert, "Investigation of Retention and Perceptions Among Freshman Engineering Students," Proceedings of the 2007 North Midwest ASEE Regional Conference, Houghton, Michigan, September, 2007.

[4] R. Hensel, J. Sigler , A. Lowery, "Breaking The Cycle Of Calculus Failure: Models of Early Math Intervention to Enhance Engineering Retention." Proceedings of the American Society for Engineering Education Annual Conference & Exposition, 2008.

[5] C. Grattis, B. Hill, A. Lachowsky, "A Successful Peer Mentoring Program," Proceedings of the American Society for Engineering Education Annual Conference & Exposition, June 2007.

[6] M. Johnson, S. Sheppard, "Students Entering and Exiting the Engineering Pipeline—Identifying Key Decision Points and Trends." Proceedings of the Frontiers in Education Conference, November 2002, pp. S3C-13 – S3C-19.

[7] D. Knight, L. Carlson, J. Sullivan, "Improving Engineering Student Retention through Hands-On, Team Based, First-Year Design Projects," Proceedings of the Conference on Research in Engineering Education, June 2007, pp. 1-13.

[8] J. Lott, S. Gardner, D. Powers, "Doctoral Student Attrition in the STEM Fields: An Exploratory Event History Analysis," Journal of College Student Retention: Research, Theory, & Practice, 2009-2010, pp. 247-266.

[9] R. Subotnik, R. Tai, R. Rickoff, J. Almarode, "Specialized Public High Schools of Science, Mathematics, and Technology and the STEM Pipeline: What Do We Know Now and What Will We Know in 5 Years?" Roeper Review, 2010, pp. 7-16.

[10] C. Ashby. Higher Education: Science, Technology, Engineering, and Mathematics Trends and the Role of Federal Programs. Testimony Before the Committee on Education and the Workforce, House of Representatives (GAO-06-702T). Washington, DC: U.S. Government Accountability Office, 2006.

[11] A. Masnick, S. Valenti, B. Cox, C. Osman, "A Multidimensional Scaling Analysis of Students' Attitudes About Science Careers," International Journal of Science Education, 2010, pp. 653-667.

[12] U.S. Department of Labor, Bureau of Labor Statistics. Occupational Outlook Handbook, 2010-2011 Edition. Washington, DC: U.S. Department of Labor, Bureau of Labor Statistics, 2010.

[13] N. Klingbeil, R. Mercer, K. Rattan, M. Raymer, D. Reynolds, "Rethinking Engineering Mathematics Education: A Model for Increased Retention, Motivation and Success in Engineering," Proceedings of the American Society for Engineering Education Annual Conference & Exposition, June 2004, pp. 12169-12180.

[14] D. Sathianathan; P. Yaeger; S. Tavener; S. Armentrout; R. Marra, "Using Applied Engineering Problems in Calculus Classes to Promote Learning," Proceedings of the Frontiers in Education Conference, November 1999, pp. 12D5-14 – 12D5-18.

[15] S. Brainard, L. Carlin, "A longitudinal study of undergraduate women in engineering and science." Proceedings of the Frontiers in Education Conference, - vol. 01, November 1997, pp 134-143.

Youth APPLab: One Response to (Lacking) Computer Science Education for Students of Color

L. Hatley[1], and A. Washington[2]

[1]Center for Learning & Educational Media, Uplift, Inc. Washington, DC, USA

[2]Department of Systems and Computer Science, Howard University, Washington, DC, USA

Abstract - This paper discusses current conditions of computer science education for students of color and highlights the benefit of several successful projects in addition to Youth APPLab, a year-long winning project funded by the MacArthur Foundation's 2010 Digital Media & Learning Competition. With proper guidance and motivation, African-American students are motivated and engaged in learning how to become software engineers. Starting with various visual programming environments, students quickly want to advance to text-based languages. These visual tools prove to be great introductions to computer programming but do not teach core concepts. Youth APPLab instructors have observed these and other project results and will use project success to offer subsequent sessions and build and test digital and paper-based tools and other teaching aids.

Keywords: Computer Science, Education, After-School, Students of Color, Mobile

1 Introduction

Youth APPLab occupies a unique place in the sizeable collection of out-of-school programs for youth in the United States. It blends instruction in computer science and software engineering with cutting edge technology and research (qualitative and quantitative inquiry) to determine perspectives of its students as they learn to create mobile applications (apps). Youth APPLab is the first and longest running program of its kind. There are 20 Youth APPLab students total, the youngest student is 8, and the older is 17 (Figure 1). All students are African-American. While current statistics defines this demographic as the underrepresented population in science, technology, engineering, and math (STEM) disciplines, Youth APPLab places Linux laptops and android mobile phones in their hands in order to determine how best to teach African-American middle and high school students how to develop software. This paper presents results from a summary of computer science education for students of color. It follows with brief descriptions of some promising programs created to expose and educate African-American students to STEM concepts. Youth APPLab is then described in detail, along with an introduction to the various methods, tools, and approaches used to teach its students. The paper ends with early findings and future work.

Figure 1. Youth APPLab students creating mobile apps.

1.1 The Need

The production of U.S. engineers and scientists is in significant decline. The extent of this decline is illustrated in the gap forming between the demand for and the amount of qualified, U.S. citizens available to fill STEM positions as the baby-boomer generation enters retirement. As the U.S. attempts to meet the needs for qualified candidates without relying on off-shore options, this gap, referred to as the "Quiet Crisis," continues to increase [4].

The emerging economies of Asia have concentrated on closing the knowledge gap between them and the western world. Although enrollment rates are lower than those in the U.S., China and India currently produce more college graduates per year, as a result of larger populations. In 2007 China produce roughly 5 million graduates [8], compared to 1.5 million in the U.S. Chinese and Indian graduates are also more heavily concentrated in STEM fields. In China, over 40% of all college graduates hold engineering degrees. In the US, that number is closer to 5%.

This “Quiet Crisis” is due to low interest in STEM for U.S.-born students [4]. According to NSF data, from 1986 to 1996 interest in science and engineering experienced a precipitous decline [8]. While by 2004, graduation rates in computer science rebounded to some extent, the graduation data for the physical sciences and engineering remains poor. In fact, interest in engineering has declined for the past 20 years and is near a 30 year low. This fact, coupled with the changing demographics of American society, makes STEM education an area of urgent national attention.

Over the next 50 years, it is projected that the minority population will increase at a much faster rate than the non-minority population, with the African-American and Hispanic workforce populations increasing from a combined 22% to 40% of the workforce [1][7]. Despite representing approximately 20% of the population in 1997, only 6% of African-Americans and Hispanics were employed in STEM-related careers. Many experts agree that, in order to fill these gaps in the workforce, more students of color, specifically African-Americans, will have to be placed in the pipeline early in their academic career. The challenge to address is how to best attract and retain these students, who are not traditionally high performers in science and math, as early as K-12.

1.2 Computer Science Education for Students of Color

Over the past five years, various research studies and project-based learning environments have been established to uncover and expand the current state of computer science education for students of color (specifically African-American and Latino). One study and four programs are presented here followed by a special emphasis on Youth APPLab, the focal point of this paper.

1.2.1 Stuck in the Shallow End

Starting in 2000, a longitudinal study began to investigate why so few African American and Latino/a high school students were studying computer science. This study resulted in the publication of a book entitled Stuck in the Shallow End: Education, Race, and Computing [6]. In it, Jane Margolis and a team of researchers described a 3-year study of various schools in the Los Angeles Public School System. They uncovered that the lack of or limited learning opportunities present in schools where the majority of student are of color as one of many reasons why students of color are not learning 'real' computer science. Margolis points to stereotypes and other challenges for these limited opportunities including: bias of teachers, guidance counselors, and administrations regarding who is 'fit' to learn computer science or has a natural inclination for it, low quality/prepared teaching staff available, and the low priority placed on ‘real’ computer classes in schools where other issues are extremely more pressing - i.e. low test scores. ‘Real’ here implies those classes that involve computer programming, simulation, research, and problem-solving as opposed to learning computer literacy skills (i.e. browsing the web and how to use software from productivity suites like Microsoft Word (cut, copy, paste, and typing). The book describes these stereotypes as being so ingrained that they were the norm and were never questioned - by anyone. The common thought was that White males and Asians are the ones who should be learning ‘real’ computer science.

2 Teaching ‘Real’ Computer Science Concepts to Students of Color

A number of multi-year, project-based, research and non-formal learning environments have been established to provide quality and equitable access to technology instruction for students of color. These projects have been extremely successful in exposing participants to the creation and presentation of game-based and digital media projects. Three are briefly mentioned below. Through various mentoring models, Clark’s Game Design Through Mentoring and Collaboration project [2], Scott’s COMPUGIRLS project [3] (both funded by the National Science Foundation), and Pinkard’s iRemix project [5] (funded by the MacArthur Foundation) have resulted in contributions towards the discovery of how African-American students develop identity through digital media, virtual worlds, and explore social justice and advanced science topics (i.e. physics) via game creation.

The projects highlighted above use game design, virtual worlds, and digital media production to introduce and engage African-American students in

STEM fields. Here, the authors introduce, Youth APPLab, a MacArthur Foundation 2010 Digital Media & Learning (Learn Lab) Competition winner. Youth APPLab is designed to teach software development skills to twenty African-American middle (8) and high school (12) students from public, charter, and home-schooled teaching environments in Washington, DC, USA.

3 Youth APPLab: A Project to Teach Students of Color how to Program Mobile Apps

Youth APPLab students learn how to use various visual programming environments (namely, Alice, Scratch, and App Inventor [9]) to create short programs, animations, and smartphone application development (android apps) in a year-long after-school environment (figure 2). Students, eight female and twelve male, meet twice a week for instruction with one additional optional open-lab meeting to allow more time to work. The course is hosted at Howard University in Washington, DC and uses the visual mobile application development environment developed by Google called App Inventor. Students are encouraged to blog (video & text) about their experiences after each class period. The project's URL is http://www.youthapplab.com. Leshell Hatley is the project's Lead Instructor. The assistant instructor is a Java software developer named Marco Jacobs.

3.1 Youth APPLab: Approaches, Methods, Motivation, Tools, and Assignments

Youth APPLab was designed to teach participants various software development concepts and provide authentic computer programming experiences. Beginning with a pre-test, classes have been conducted for the past seven months. Instructors have applied various methods and approaches to teaching, initiated several motivational techniques, and constructed various learning tools to aid in content mastery.

3.1.1 Approaches and Methods

Lessons began with broad discussions about technology in general, they then moved toward the targeted platform, mobile devices. This grew into immersive uses of this these devices as each student was given his/her own android phone with unlimited talk, text, and web connectivity features. Lectures were given; homework assigned, and class work was done by instructors and students together. These classroom activities were accompanied with example spotlights of other young mobile application developers as well as

Figure 2. Female Youth APPLab student working on Scratch program.

the staggering statistics of low STEM participation by people of color. These activities were followed by open-ended discussions about the content, learning processes, and thoughts about future participation in computer science and other STEM related fields.

3.1.2 Assignments and Tools

Instructions are given in traditional instructor-to-classroom lecture style along with one-on-one assistance, and assignments that are done together in class. Computer Programming lessons began with Alice, moved to Scratch, and then ultimately to App Inventor – all visual programming environment designed for computer programming novices [9]. Class discussions continue via email and/or within group online chat sessions. Instructors and students work together to develop various learning tools and approaches to ensuring that content is mastered.

3.1.3 Motivation

The course also employs various motivational and encouragement techniques. These are especially employed when student are discouraged when facing a new task, one never experienced before, and/or when they have tried the same task several times with little success.

3.2 Youth APPLab as Research

The goal of Youth APPLab goes beyond exposing African-American youth to 'real' computer science and cutting-edge technology. It allows students the opportunity to innovate, problem-solve, communicate, and express themselves. As such, Youth APPLab is also a vehicle for probing students' thoughts and perspectives about technology and the best approaches to learning computer science and software development.

3.3 Early Findings

African-American students of all ages are extremely engaged in the activities of a software engineer and, with guidance, are eager to learn more and move on to advanced levels. At the onset of Youth APPLab, 85% of participating students did not know what "programming" and/or "coding" meant. Currently, 100% are designing and programming 1 or 2 apps of their own creation. All students want some form of computer programming offered as a class during the standard school day and many are now moving towards majoring or minoring in computer science or information technology when they attend college (figure 3).

One 15-year old male student says this about his participation in Youth APPLab: "*I thought it'd be a good opportunity to enhance my knowledge of computers, while getting a head start on a possible career. I'm glad to say that I have not been disappointed*." Many students realized that the Software Development Life-Cycle (SDLC) can be used and applied to most school projects. One 17 -year old male student expressed an interest in moving on to advanced levels of programming – beyond the visual environments used in class. "*I want to be at a blank screen and do this*," (while moving his fingers as if typing). After one month of instruction, he and many other students wanted to do more than drag and drop, the typical programming behavior of visual programming environments.

Other students revealed various challenges with learning some computer programming concepts. For example, while learning about algorithms, one 15-year old female student was under the impression that you had to include what the programmer actually does when creating algorithms. She expressed the beginning steps of her algorithm as "*programmer opens Scratch and begins to drag wanted puzzle pieces onto the Stage*." Another 15-year old male student explains, "*The concepts aren't hard to grasp at all. The hard part is explaining them. I [get] it, I just don't know how to say it*."

These are just a few thoughts shared by students regarding their participation in Youth APPLab. More advanced investigation, both qualitative and quantitative, will take place upon the project's completion.

Figure 3. Youth APPLab students and instructor.

4 Conclusions and Future Work

Youth APPLab instructors have found that the visual programming environments used in class do not teach some fundamental computer science concepts, i.e. what an algorithm is, how to write one, the steps towards problem-solving, programming constructs, and many others.

From these early findings and others throughout the remaining portion of the project, Uplift, Inc. and its Center for Learning and Educational Media seek to create additional tools (software and paper-based) that can be used to teach these and other computer programming and software engineering concepts more efficiently. Within the next several months, more in-depth qualitative and quantitative analysis will be conducted on the content knowledge through the analysis of pre- and post-test scores, students' perspectives on STEM and Computer Science, application (app) designs, and code quality and complexity. Parent feedback will also be solicited and evaluated.

Subsequent Youth APPLab sessions will be given using the created tools from this initial round to investigate the effectiveness of tools and approaches used. Replication efforts will also commence to aid in the adoption of this class by other out-of-school programs serving the same and other populations.

5 Acknowledgements

We gratefully acknowledge the financial support of the MacArthur Foundation and their annual Digital Media & Learning Competition. We also acknowledge HASTAC, the administrators of the competition and ultimate awards. We especially thank each of our students for their hard work and persistence. We also thank Howard University for hosting our classes and each parent for allowing his/her child to participate. We also thank our staff, volunteers, and many guests for their interest, time, and attention.

6 References

[1] Chubin, D., May, G., and Babco, E. “Diversifying the Engineering Workforce.” Journal of Engineering Education. Jan, 2005.

[2] Clark, K. and Sheridan, K. (2010). Game design through mentoring and collaboration. Journal of Educational Multimedia and Hypermedia, 19(2), pp. 125-145.

[3] COMPUGIRLS, http://sst.clas.asu.edu/compugirls

[4] Jackson, S.A. “The Quiet Crisis: Falling Short in Producing American Scientific and Technical Talent.”Building Engineering and Science Talent. www.bestworkforce.org/ODFdocs/Quiet_Crisis.pdf

[5] Kennedy Martin, C, Barron, B, Austin, K and Pinkard, N. “A Culture of Sharing: A Look at Identity Development Through the Creation and Presentation of Digital Media Projects.” International Conference on Computer-Supported Education (Lisbon, Spain) March. (2009).

[6] Margolis, J. *Stuck in the Shallow End: Education, Race, and Computing*. The MIT Press, 2008.

[7] National Academy of Engineers. “The Engineer of 2020 Project Prospectus”. http://www.nae.edu/nae/engeducom.nsf/weblinks/MCAA-5L3MNK?OpenDocument

[8] National Science Board “An Emerging and Critical Problem of the Science and Engineering Labor Force” (2004). http://www.nsf.gov/nsb/

[9] Visual Programming Environments designed for Novice Programmers: Alice: http://www.alice.org; Scratch: http://scratch.mit.edu’ App Inventor: http://appinventor.googlelabs.com

SESSION

INTERESTING STUDIES + LEARNING METHODS AND TECHNIQUES + TUTORING + RELATED ISSUES

Chair(s)

TBA

Visualizing Logical Thinking using Homotopy A new learning method to survive in dynamically changing cyberworlds

Kenji Ohmori[1], **Tosiyasu L. Kunii**[2]
[1]Computer and Information Sciences, Hosei University, Koganei, Tokyo, Japan
[2]Morpho Inc, Bunkyo, Tokyo, Japan

Abstract— *For logical thinking learners, it is important to learn problem solving formally and intuitively. Homotopy which is modern mathematics combines algebra and geometry together. Homotopy gives the most abstract level of explanation or the most basic invariant which is the most required characteristics for logical thinking. The homotopy lifting property gives good explanation of inductive reasoning. The homotopy extension property, which is dual to the homotopy lifting property, provides a good tool for deductive reasoning. As homotopy combining algebra with geometry, it provides intuitive explanation with informative figures. This paper shows how puzzle-based learning is explained mathematically and intuitively using these properties.*

Keywords: A maximum of 6 keywords
logical thinking; puzzle-based learning; inductive and deductive reasoning; homotopy lifting and extension properties; problem-solving

1. Introduction

Logical thinking [1], [2], [3] is an important method to solve inexperienced problems encountered in the complicated and sophisticated society of cyberworlds, where we are insisted to survive wisely in dynamically and rapidly changing surroundings. Logical thinking needs mathematics to solve problems formally and logically. However, learners are awkward to use classical mathematics which uses many expressions so that symbol manipulation is not intuitively understandable.

Graphical user interface, which is improved with the development of Web systems, provides a visualized environment for logical thinking tools. Homotopy [4], [5], [6] in the field of topology of modern mathematics gives a better way for visualizing logical thinking since homotopy unifies algebra and geometry with informative illustrations of figures.

In cyberworlds, homotopy has been applied to software development [7], [8], [9], where system requirements are formally transformed into concurrent processes described by the pi-calculus and then implemented as event-driven and multi-thread programs suitable for XMOS processors. The paper expresses that homotopy has two important properties: the homotopy lifting and extension properties. Furthermore, when applying homotopy into the field of cyberworlds, these two properties are used as tools for bottom-up and top-down methods, respectively.

Logical thinking is mainly employed by deductive reasoning and inductive reasoning. Inductive reasoning is a thinking skill that is initiated by observing something, and then completed by a conclusion based on what has been observed. Inductive reasoning is thought as the process of inducing an abstract property from specific properties since it leads a common property or an invariant from specific facts. Deductive reasoning is a thinking skill that is started by forming a conclusion, and then achieved by showing that the conclusion follows from a set of premises. Deductive reasoning is considered as the process of deducting an abstract property into specific properties, so that inductive and deductive reasoning are equivalent to the bottom-up and top-down methods, respectively, used in the software development application.

The homotopy lifting property and extension properties can be used as intuitive and informative skills since these properties are rich in graphical user interface. In this paper, these properties are applied to puzzle-based learning, which is a new teaching methodology focused on the development of problem-solving skills for logical thinking. Puzzle-based learning [10] is introduced to overcome the disadvantage of the conventional problem-solving where logical thinking skills are taught after learning professional knowledge. However, students want to improve their logical thinking skills before learning professional knowledge. As puzzles can be solved without professional knowledge, it is thought that puzzle-based learning is suitable for improving logical thinking skills by solving puzzles instead of professional problems.

There may be objections to use homotopy when solving puzzles. As topology and homotopy are taught after classical mathematics, students tend to think that homotopy is a hard subject. However, as it is shown in this paper, homotopy is not a so difficult subject as compared with calculus which is taught in high schools.

2. How to employ homotopy in cyberworlds

Homotopy is defined in mathematics as follows.

[Homotopy] A homotopy between two continuous functions f and g from a topological space X to a topological space Y is defined to be a continuous function $H : X \times [0, 1] \to Y$ from the product of the space X with the unit interval $[0, 1]$ to Y such that, if $x \in X$ then $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$. If the second parameter of H is considered as time then H describes a continuous deformation of f into g where we have the function f at time 0 and the function g at time 1.

A continuous function, which appears in the previous sentence, is important concept in mathematics. It is defined as follows.

[Continuous function] A function $f : X \to Y$, where X and Y are topological spaces, is continuous if and only if the inverse image $f^{(-1)}(V) = \{x \in X | f(x) \in V\}$ is open for every open set $V \subseteq Y$.

As cyberworlds are often employed in the set of discrete elements, mathematical terms have to be interpreted to adjust discrete characteristics of cyberworlds. The set of discrete elements is transformed into a topological space by introducing the trivial space or the discrete space. When the unit interval is discrete, the elements of the unit interval is treated as the components of a totally ordered set.

Fig. 1 is an example of homotopy in cyberworlds. X and Y are the sets of discrete elements. Y is divided into the subset of $\{Y_0, Y_1, Y_2, Y_3, Y_4\}$. It is considered that X is mapped to Y_i at time t_i. This example can be used to describe how a certain area has changed in history or how the financial state of a company has changed in recent years. Now, we will introduce topological spaces to X and Y and define a homotopy H. The topological space (X, T_x) is defined by introducing the subset $T_x = \{\phi, X\} of X$. The interval is defined as follows. $t_i \in I, T_I = \{[t_i, t_j] | t_i, t_j \in I, i \leq j\}$. I is a totally ordered set. The topological space (Y, T_y) is also defined by introducing the power set T_y of $\{Y_0, Y_1, Y_2, Y_3, Y_4\}$. Then, the homotopy H is defined by $H(x, t_i) : x \in X, t_i \in I \to y \in Y$.

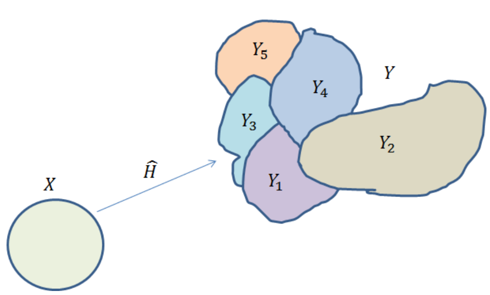


Fig. 1: Homotopy employed in cyberworlds.

3. Homotopy lifting property in a discrete space

The homotopy lifting property is used as a tool for deductive reasoning.

[Homotopy lifting homotopy] The homotopy lifting property is defined as follows. Given any commutative diagram of continuous maps as shown in Fig. 2, the map $p : E \to B$ has the homotopy lifting property if there is a continuous map $\hat{H} : Y \times I \to E$ such that $p \circ \hat{H} = H$. The homotopy $\hat{H}$ thus lifts H through p.

Fig. 2 also includes an example of the homotopy lifting property. E is the surface of a grove and B is the projected space of B. X is a line which is mapped to E. However, the mapped lines are continuously changed along the time interval I as shown in Fig. 2.

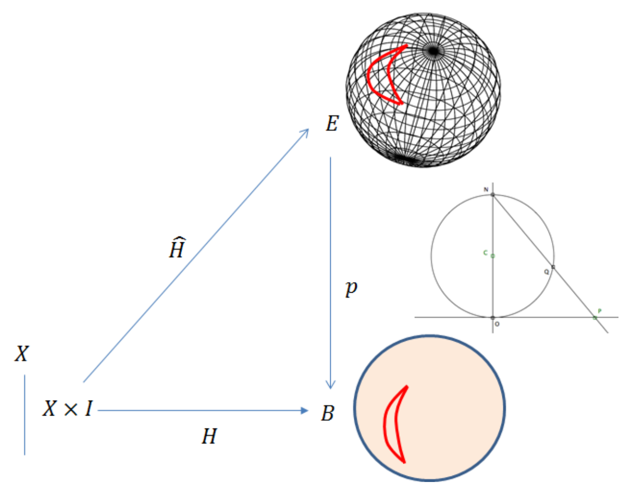


Fig. 2: Homotopy Lifting Property

[Fiber bundle] A fiber bundle is a quadruple $\xi = (E, B, F, p)$ consisting of a total space E, a base space B, a fiber F, and a bundle projection that is a continuous surjection called $F-$bundle $p : E \to B$ such that there exists an open covering $\mathcal{U} = \{U\}$ of B and, for each $U \in \mathcal{U}$, a homeomorphism called a coordinate chart $\varphi_U : U \times F \to p^{-1}(U)$ exists such that the composite $U \times F \to p^{-1}(U) \to U$ is the projection to the first factor U. Thus the bundle projection $p : E \to B$ and the projection $p_B : B \times F \to B$ are locally equivalent. The fiber over $b \in B$ is defined to be equal to $p^{-1}(b)$, and it is noted that F is homeomorphic to $p^{-1}(b)$ for every $b \in B$, namely $\forall b \in B, F \cong p^{-1}(b)$.

A Mobius strip gives a good example to explain a fiber bundle as shown in Fig. 3. If a Mobius strip and its center circle described by the dot line are considered as E and B of a fiber bundle and any point of E is projected to the intersection of the center circle with the perpendicular line from the point of E to the center circle, then, F is a straight line segment vertical to the center circle.

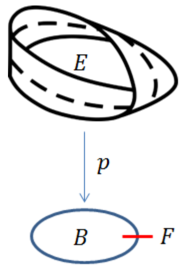


Fig. 3: Mobius strip.

Though a cylinder is different from a Mobius strip in shape, these objects are alike in homotopy. When constructing a fiber bundle for a cylinder, the cylinder has the same B and H as a Mobius strip. B of the cylinder is also a circle. The fibers are also straight line segments vertical to the center circle. However, the fibers of the Mobius strip is rotating along the center circle. On the other hand, the fibers of the cylinder are always straight, which causes the difference in shape. B is considered as a common property or invariant among similar objects in homotopy. A cylinder is deformed to the circle by moving any point perpendicularity to the point of the center circle. A Mobius strip is also deformed to the circle by the same moving. In the abstract level, a cylinder and a Mobius strip are the same. The invariance of two objects is explicitly represented by B.

Google map is a good example to explain how a fiber bundle is applied in cyberworlds as shown in Fig. 4. E is a Google map which is constructed by a set of guide map layers, each of which corresponds to a business category such as restaurants, hospitals, and so on. B is the projected map from E. B is the common property of the layers and invariant.

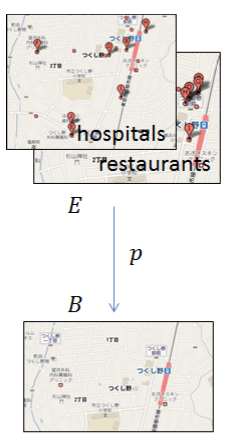


Fig. 4: Google map.

4. Inductive reasoning using homotopy lifting property

The homotopy lifting property explains mathematically and intuitively how a puzzle is solved by inductive reasoning. Let us consider to solve the following puzzle.

[Gas station puzzle] There are n gas stations located on a circular track. The total gas available in these gas stations is just enough for one car to complete the loop. Show that there is a gas station from which a car (initially with an empty tank) can start and complete the loop by collecting gas from the gas station on its way around. (Z. Michalewicz, M. Michalewicz, Puzzle-Based Learning: An introduction to critical thinking, mathematics, and problem solving, p64)

The puzzle is specified by providing 12 gas stations, each of which is given a name, distance from station A and available gas as shown in Table 1.

Table 1: An Example of a Table

St	A	B	C	D	E	F	G	H	I	J	K	L
Dis	0	30	80	115	150	170	205	215	245	295	305	325
Qt	9	5	10	8	4	1	7	4	4	12	3	5

The commutative diagram of the homotopy lifting property is constructed as shown in Fig. 5. The information of gas stations provided in Table 1 is assigned in X. X is the list of stations with name, distance and available gas. That is, $x = ((A, 0, 9), (B, 30, 5), ...) \in X$. I is a series of the starting points, from each of which the car starts to go around the circular track. That is $I = (t_0, t_1, ..., t_{11})$, where t_i means that the car starts from the $(i + 1)^{th}$ station. The element of $X \times I$ is a pair of the gas station list and a starting point. $H(x, t)$ maps the station list from starting point t to a remaining gas graph. E is constructed to show how much gas remains at the tank along the circular track starting from the gas station designated by I. When gas is exhausted, E shows how much gas should be borrowed.

In Fig. 5, the graph for t_i shows the change of remaining gas along the track when the car starts from the $(i + 1)^{th}$ station. If remaining gas is negative, the same amount of gas must be borrowed. The graphs on E are different in quantity. But, these graphs have the same shape, which is a common property of these graphs. By removing the values of remaining gas and connecting the starting station with the ending so that a graph becomes a loop, each graphs on E is projected to B as shown in Fig. 5. As B is invariant in this puzzle, we can obtain the solution of the puzzle from B. Since B is a loop, it has at least one lowest point as an invariant. If the car starts from the lowest position, the car can run the circular track without running out of gas.

There is another puzzle, which is solved using mathematical induction.

[Strange country] There is a strange country where every road is a one way road. Moreover, every pair of cities is connected by only one directed road. Show that there exists

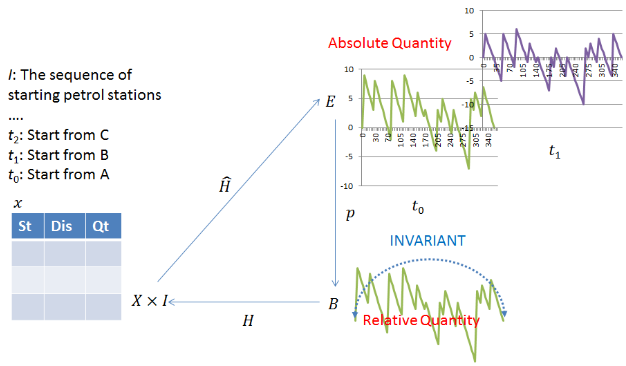


Fig. 5: Homotopy lifting property for gas station puzzle.

a city (possibly the capital of this strange country?) that can be reached from every city directly or via at most one other city. (Z. Michalewicz, M. Michalewicz, Puzzle-Based Learning: An introduction to critical thinking, mathematics, and problem solving, p66)

By assuming that the capital exists for n cities, let's try to prove that the capital exists for $n+1$ cities. This puzzle is usually proved by mathematical induction. Mathematical induction can be visualized using the homotopy lifting property. At first, let's define X and I. A new country with $n+1$ city is constructed by providing the new roads that connect a new city with the old cities. As a road is one way, the number of countries with $n+1$ cities becomes the combination of one way roads. The one way roads connecting the new city with the old cities are described by $(c_1, c_2, ..., c_n)$ where $c_i = 0$ if the road is directed from the i^{th} city to the new city, otherwise $c_i = 1$. I is defined by the combination of directions and is sorted as Huffman codes. The set of directions is described by $I = (t_0, t_1, ..., t_{2^n-1})$ and $t_0 = (0, 0....0, 0)$, $t_1 = (0, 0....0, 1)$, $t_2 = (0, 0....1, 1)$,.... $t_{2^n-1} = (1, 1....1, 1)$. X is constructed as the country with n cities. By assumption, n cities are classified into three groups as shown in Fig. 6: the capital, the directly reachable cities from which a one-way road to the capital is provided, and others, which are named indirectly reachable cities. From an indirectly reachable city, a one-way road to a directly reachable city is provided. Homotopy $\hat{H}$ is defined by $\hat{H}(x, t) = y$ where $x \in X, t = (c_1^t, c_2^t, ..., c_n^t) \in I, y \in E$.

The countries with $n+1$ cities can be divided into three layers: L_0, L_1, L_2 according to how new roads connected to the new added city are provided. If the roads are provided so that the capital is directly reached from the new city, these countries belong to L_0. If the roads are provided so that the new city is reached from the capital and at least one of the directly reachable cities are reached from the new city, these countries become L_1. Otherwise, the countries belong to L_2 where an old indirectly reachable cities become a directly or indirectly reachable city. The old capital remains as the capital at L_0, L_1 and the new city becomes the new capital at L_2. The countries in E is projected to the graph of B, which is invariant. The graph of B is the same as one of X. Therefore, the puzzle has been proved.

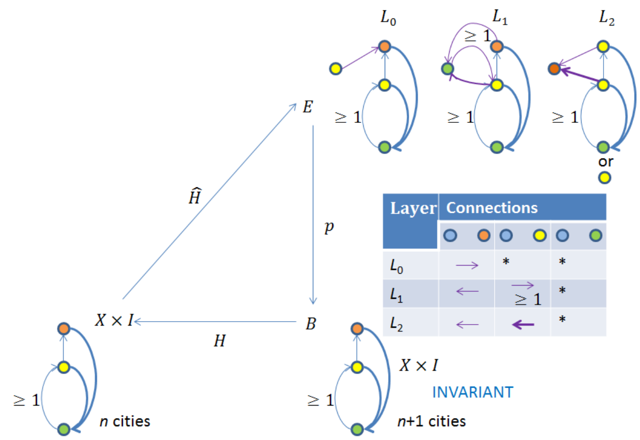


Fig. 6: Strange country

5. Homotopy extension property in a discrete space

The homotopy extension property is dual to the homotopy lifting property, which is defined as follows.

[Homotopy extension property] Given any commutative diagram of continuous maps as shown in Fig. 7, there is a continuous map $\hat{K} : X \to Y^I$ such that $p_0 \times \hat{K} = k$ and $\hat{K} \times i = K$. The homotopy $\hat{K}$ thus extends K over i and lifts k through p_0 where $p_0(\lambda) = \lambda(0)$.

In the above definition, λ is called a path and is defined as follows.

[path] A continuous map $\lambda : I \to X$ yields a path. $\lambda(0) = x$ and $\lambda(1) = y$ are called the initial and terminal points. The path is denoted by $w = (W; \lambda)$ where $W = \lambda(I)$.

Y^I is called a path space and is defined as follows.

[Path space] The path space on Y, denoted Y^I , is the space $\{\lambda : I \to Y | continuous\}$.

In Fig. 7, an example of the homotopy extension property in cyberworlds is depicted. The example shows a Google map application where a Google map provided for each customer is generated from the Google map. X is the Google map consisting of business category layers, each of which shows the places of buildings for a given business category, such as restaurants, hospitals and so on. A is the map for those layers which is included in X. Y^I is a set of Google maps for each customer. I is a series of categories. A set of continuous functions $\{\lambda\}$ is assigned for each customer such as Peter, Betty and so on. $\lambda_p(I)$ is a path for Peter, which describes a Google map for Peter.

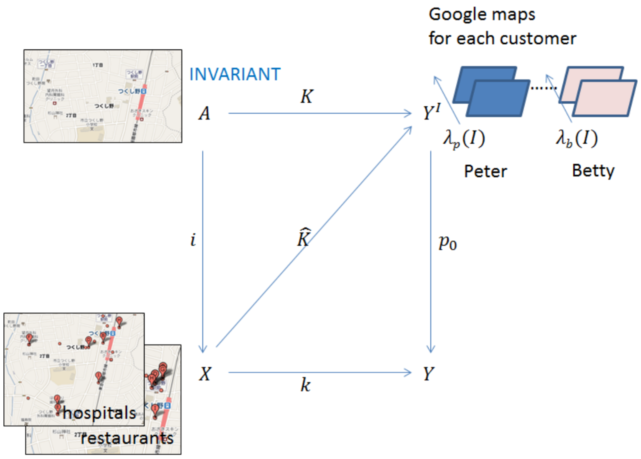


Fig. 7: Homotopy Extension Property

6. Deductive reasoning using homotopy extension property

The homotopy extension property explains intuitively how a puzzle is solved by deductive reasoning. Let us consider to solve the following puzzle.

[Chess board] There is a chess board with a dimension of 2^m(i.e., it consists of 2^m squares) with a hole, i.e., one arbitrary square is removed. There are number of L shaped tiles and your task is cover the board with these tiles (the orientation of a tile is not important). How can you do it? Where should you start? (Z. Michalewicz, M. Michalewicz, Puzzle-Based Learning: An introduction to critical thinking, mathematics, and problem solving, p68)

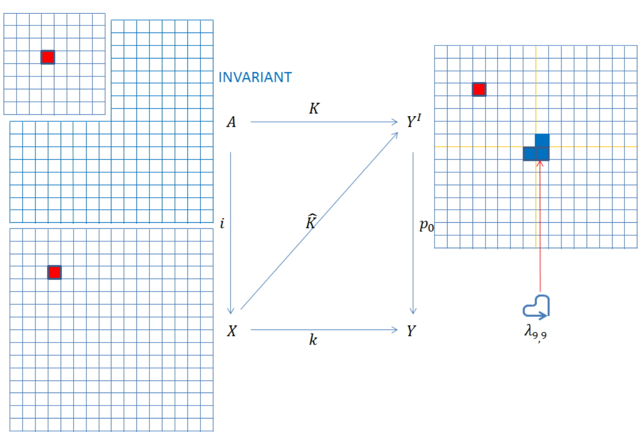


Fig. 8: Chess board puzzle - Divide and Conquer

The puzzle is solved using the homotopy extension property as shown in Fig. 8. To simplify the puzzle, a chess board with 16×16 squares is provided, so that 85 L shaped tiles have to be placed on the chess board.

The commutative diagram of the homotopy extension property is constructed as follows. X is assigned as the chess board where a hole is arbitrary provided. A is part of the chess board. The chess board is divided into four equal rectangular blocks and the block which has the hole is detached from the other blocks. The other blocks form an L shaped board. Therefore, A consists of the rectangular block with the hole and the L shaped board without the hole. I is a sequence of times at which an L shaped tile is placed on the chess board. That is, $I = (t_0, t_1, ..., t_{85})$ At t_i, the i^{th} tile is placed. Y^I is a path space that describes the changes of the chess board by placing L shaped tiles.

Now, let'us consider to place the first L shaped tile on the chess board. The rectangular square is possible to be covered by L shaped tiles since the number of squares that have to be covered by tiles can be divided by three. (Three is the number of squares that a tile needs to occupy.) Other blocks that does not have the hole become possible to be covered by tiles if one square is removed from each block. It is achieved by placing an L shaped tile on the three blocks, so that it surrounds the corner where these blocks meet together.

As Y^I is a path space, paths must be defined. Let's define the elements of X and Y^I as follows. $X = \{b_{i,j} | i, j = 1..16\}$ where i and j are a row and column number, respectively. Y is the same as X. $Y^I = \{\lambda_{i,j}(t) | i, j = 1..16, t \in I\}$ where $\lambda_{i,j}$ starts from $b'_{i,j} \in Y$. Homotopy $\hat{K}$ is defined by $\hat{K}(b_{i,j}, t) : b_{i,j} \in X, t \in I \rightarrow \lambda_{i,j}(t) \in Y^I$.

When the first tile is placed, path $\lambda_{9,9}(t)$ starts from $b'_{9,9}$ at time t_0, circles around the tile and ends at $b'_{9,9}$ at time t_1. At the same time, $\lambda_{8,9}(t)$ and $\lambda_{9,8}(t)$ circulate around the tile.

This process is repeated to each divided four blocks. Fig. 9 shows how a new L shaped tile is placed in the divided block with the hole. For constructing Y^I, other pathes $\lambda_(5,5)$, $\lambda_(4,5)$ and $\lambda_(5,4)$ starting at t_1 and ending at t_2 are provided by circling around the new tile.

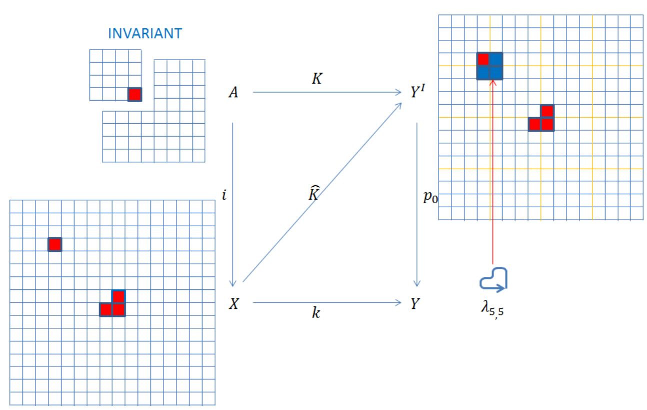


Fig. 9: Chess board puzzle - Divide and Conquer

7. Conclusions

The paper describes how homotopy is applied to logical thinking. The first puzzle uses the homotopy lifting property for inductive reasoning. The invariant shape showing the change of remaining or borrowing gas has been obtained by projecting the specific patterns of E to the generalized pattern to B. The process is mathematically explained by the commutative diagram of the homotopy lifting property how the information of gas stations in X is mapped to the graph of remaining or borrowing gas in E according to the starting gas station of I using informative figures. The figures of E are different in absolute values but the common shape is projected into B, which is invariant and gives an answer to this puzzle.

The second puzzle is a problem of mathematical induction. The puzzle also uses the homotopy lifting property in a bottom-up way as a tool for inductive reasoning. It is assumed that the condition is satisfied for the case of n and has to be proved for the case of $n+1$. The case of n is described in X, where all situations are not represented as graphs, but a typical graph that is equivalent to the all graphs is depicted in X. I is constructed as a sequence of cases for forming one way roads between the old new cities and a new city, which constitutes the case for $n+1$. The new situations in E are classified into three layers, which are mapped to B to obtain a common property of the $n+1$ case. As the problem-solving process is explained structurally by the diagram and intuitively using graphs, the homotopy application to the second puzzle has succeeded in obtaining a good tool for logical thinking.

The third puzzle is a divide-and-conquer problem. The problem is solved by deductive reasoning. Therefore, the homotopy extension property is used in a top-down way. The homotopy extension property is dual to the homotopy lifting property. However, as it uses a path space, which is unfamiliar concept, it is considered that the homotopy extension property is hard to be applied in cyberworlds. The third puzzle has succeeded in not only solving a divide-and-conquer problem but also expanding how a path space is usefully applied to cyberworlds.

Through solving these problems, we have succeeded in introducing homotopy as a tool of logical thinking, which leads to improve skills of logical thinking sophisticatedly.

References

[1] H. W. Dettmer, *The Logical Thinking Process: A Systems Approach to Complex Problem Solving*. Milwaukee, Wisconsin: ASQ Quality Press, 2007.
[2] D. McInerny, *Being Logical: A Guide to Good Thinking*. New York: Random House, 2004.
[3] P. George, *How To Solve It*. Princeton, New Jersey: Princeton University Press, 2004.
[4] C. Dodson and P. E. Parker, *A User's Guide to Algebraic Topology*. Boston: Kluwer Academic Pub, 1997.
[5] A. J. Sieradski, *An introduction to topology and homotopy*. Boston: PWS-Kent Publishing Company, 1992.
[6] E. H. Spanier, *Algebraic topology*. New York: Springer-Verlag, 1966.
[7] K. Ohmori and T. L. Kunii, "A pi-calculus modeling method for cyberworlds systems using the duality between a fibration and a cofibration," *Int. Conf. on Cyberworlds 2008*, pp. 363–370, September 2008.
[8] ——, "Mathematical foundation for designing and modeling cyberworlds," *Int. Conf. on Cyberworlds 2009*, pp. 80–87, September 2009.
[9] ——, "Designing and modeling cyberworlds using the incrementally modular abstraction hierarchy based on homotopy theory," *The Visual Computer*, vol. 26, no. 5, pp. 297–309, May 2010.
[10] Z. Michalewicz and M. Michalewicz, *Puzzle-based Learning: Introduction to critical thinking, mathematics, and problem solving*. Melbourne: Hybrid Publishers, 2008.

The Effect of Computer Programming Education on the Reasoning Skills of High School Students

Robert W. Fox* and Michael E. Farmer+
* Midland Public Schools, Midland, MI, USA
+Department of Computer Science, Engineering Science, and Physics
University of Michigan-Flint, Flint, MI, USA

Abstract-*The goal of this study was to determine if high school students learning computer programming would demonstrate improved reasoning skills. To test this, sixty 11th and 12th grade students were sampled from a high school population and half of these students were treated with one semester (75 hours) of computer programming instruction while the other half was not. After treatment, both groups were given a standardized reasoning skills inventory that covered deductive reasoning, inductive reasoning, credibility and observational reasoning, reasoning about assumptions, and reasoning about meaning. Results were mixed, but did reveal higher mean scores in the experimental group than the control group in terms of overall and most specific areas of reasoning. In particular, deductive and credibility reasoning showed the most promise in terms of being improved by learning computer programming.*

Keywords - Computer Science Education, Cognitive Reasoning, Logical Reasoning.

1. Introduction

The ability to reason about a situation in order to solve a problem or plan a course of action is a thinking skill that is inherently important to functioning in human society. The importance of reasoning cannot be overstated considering the fact that every decision a human makes is based on the analysis of existing knowledge. Among other things, this analysis is used to speculate outcomes, weigh alternatives, assess risks, estimate rewards, and eventually draw conclusions which may be acted upon. Indeed, there has been a great deal of focus in the educational field towards engaging students in higher order thinking skills such as reasoning and problem solving. This is largely due to the recognition that learning factual knowledge does not imply the ability to use that knowledge in a meaningful way.

Computer programming is an activity that necessarily involves problem solving as the description of a program to be written is, itself, a problem to be solved using the computer as a tool. A course in computer programming, therefore, provides a student with a great deal of sustained engagement in problem solving and reasoning. Certainly, successfully completing a computer programming course will result in improved computer programming skills, but it is possible that this sustained engagement in problem solving required by the course will have an impact on the student's overall reasoning and problem-solving ability. The goal of this research is to determine if successfully completing an introductory course in computer programming will result in improved reasoning and problem-solving skills for a high school student. In order to make this determination, two groups of students will have their reasoning skills assessed using a standardized test. The experimental group will consist of thirty high school juniors and seniors who have successfully completed the course *Computer Programming 1*, and the control group will consist of thirty juniors and seniors who have not attempted this course. The results of the two groups will then be statistically compared in order to determine if, in fact, taking the introductory programming course results in improved reasoning skills.

2. Related work

Much of the research in this area has revolved around elementary aged students and was primarily done in the 1980s. These studies utilized the LOGO programming language which was a popular teaching tool in elementary schools during this time period. These studies often had conflicting results, but reported benefits, while varied, included improved divergent and creative thinking, reduced error rates on the measurement units, reduced cognitive latency, improved reflectivity, higher ability in meta-cognition, improved planning ability, improved ability to deal with multiple aspects of a problem, and increased spatial geometry skills [2] [3] [5] [7][8]. While the results of these studies add support to the research hypothesis, they were heavily criticized. These criticisms suggested that sample sizes were often too small, not randomly sampled, treatment times were too short, and that treatments were inconsistent and poorly documented making replication difficult [1][10]. Additionally it was proposed that the reported benefits of programming instruction related more to subject matter of the programming activities as opposed to the programming itself [5]. For example, activities in geometry might build geometric reasoning skills with or without computer programming. Needless to say, transferring these elementary studies' results to high school students is problematic due to the vast differences in cognitive development.

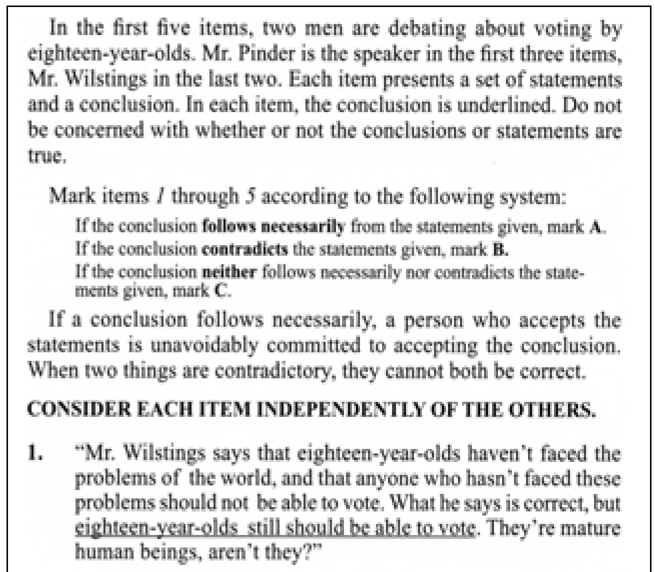
Studies examining high school and college students are rare, but a few results do exist. Research has also been mixed in terms of the benefits of computer programming instruction on cognitive skills. Though the sample size was relatively small, one study found that high school students treated with computer programming instruction systematically scored better on a standardized problem-solving test as treatment progressed compared with minimal improvement from a control group [9]. A different study conducted on undergraduate introductory programming students found more inconclusive results, but suggested that there was enough compelling evidence to warrant further study [11]. In particular the study recommended that a longer treatment time and treatment that emphasizes lab-based and hands-on learning as opposed to a lecture and demonstration style might yield more gains in problem solving for students [11]. This conclusion, along with those of the elementary studies, underscores the idea that appropriate instructional strategies that emphasize problem solving may be needed for the cognitive benefits of programming to be realized.

An interesting study was performed by Holvikivi in [12] where they discovered that first year engineering students in college lacked basic mathematical reasoning skills that faculty had believed should be present at that point in their education. It is hoped that since a significant element of programming is developing logic for dictating the program flow, that direct exposure to programming will have a more direct impact on students' logical reasoning.

3. Elements of reasoning

The concept of reasoning or critical thinking, though complex, can be thought of as "reasonably and reflectively deciding what to believe or do" [4]. Traditionally types of reasoning have fallen into a few well-known categories. These include deductive, inductive, and abductive reasoning. From these basic types, further more situational-specific types of reasoning can be derived.

The Cornell Critical Thinking Test (CCTT) was the measurement tool used in this study. It is 52-item multiple choice assessment designed to measure critical thinking abilities primarily in terms of deductive reasoning (twenty-four questions) and inductive reasoning (eighteen questions) reasoning as a strategy to infer a conclusion based upon the information provided in each problem. Other situation-specific forms of reasoning also exist in several of the questions including reasoning about observations and credibility (four questions), reasoning about assumptions (ten questions), and reasoning about meaning (fifteen questions). It should be noted that there is some overlap between the questions in terms of the specific variety of reasoning being tested. For example, all of the questions dealing with assumptions are also deductive reasoning questions.


In the first five items, two men are debating about voting by eighteen-year-olds. Mr. Pinder is the speaker in the first three items, Mr. Wilstings in the last two. Each item presents a set of statements and a conclusion. In each item, the conclusion is underlined. Do not be concerned with whether or not the conclusions or statements are true.

Mark items *1* through *5* according to the following system:

If the conclusion **follows necessarily** from the statements given, mark **A**.
If the conclusion **contradicts** the statements given, mark **B**.
If the conclusion **neither** follows necessarily nor contradicts the statements given, mark **C**.

If a conclusion follows necessarily, a person who accepts the statements is unavoidably committed to accepting the conclusion. When two things are contradictory, they cannot both be correct.

CONSIDER EACH ITEM INDEPENDENTLY OF THE OTHERS.

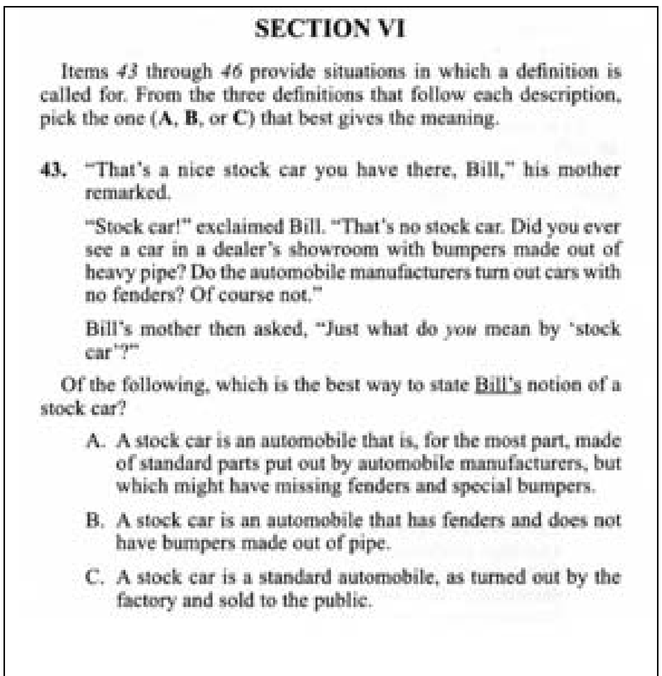
1. "Mr. Wilstings says that eighteen-year-olds haven't faced the problems of the world, and that anyone who hasn't faced these problems should not be able to vote. What he says is correct, but eighteen-year-olds still should be able to vote. They're mature human beings, aren't they?"

Figure 1: A sample question from the CCTT-Z testing deductive reasoning.

The CCTT-Z assesses reasoning by asking test takers to reason about everyday situations. Figure 1 provides a sample question from the CCTT-Z testing, in this case, deductive reasoning. For this question, the correct answer would be "B" since the conclusion drawn directly contradicts the statements given. Figure 2 provides a sample question dealing with reasoning about assumptions and meaning. In this question, test takers are asked to look at the statements made and determine the meaning of the phrase "stock car." The correct answer to this question is C, and arriving at this answer requires the test taker to determine the meaning based on the assumptions of the characters in the story.

4. Study design

This study will utilize sixty 11th - and 12th - grade male and female high school students all attending the same school. Ages of participants will range from sixteen to eighteen years. The participants will be split into experimental and control groups with thirty participants in each. To be included in the experimental group, a participant must have successfully completed the high school's introductory course in computer programming. Successful completion is determined by a final course percentage grade of 80% or higher. To be included in the control group, a participant must not have attempted the high school's introductory computer programming course. The school that the participants attend tracks students into three categories based upon mathematical reasoning ability. These tracks include a standard ability, accelerated, and honors curriculum. To ensure that both groups have a similar background in terms of reasoning ability and experience, the number of participants from each math track will be equal between the two groups. Because of this, current enrollment in a math course was also an inclusion requirement.



SECTION VI

Items *43* through *46* provide situations in which a definition is called for. From the three definitions that follow each description, pick the one (**A**, **B**, or **C**) that best gives the meaning.

43. "That's a nice stock car you have there, Bill," his mother remarked.

"Stock car!" exclaimed Bill. "That's no stock car. Did you ever see a car in a dealer's showroom with bumpers made out of heavy pipe? Do the automobile manufacturers turn out cars with no fenders? Of course not."

Bill's mother then asked, "Just what do *you* mean by 'stock car'?"

Of the following, which is the best way to state Bill's notion of a stock car?

A. A stock car is an automobile that is, for the most part, made of standard parts put out by automobile manufacturers, but which might have missing fenders and special bumpers.

B. A stock car is an automobile that has fenders and does not have bumpers made out of pipe.

C. A stock car is a standard automobile, as turned out by the factory and sold to the public.

Figure 2: A sample question from the CCTT-Z testing reasoning about assumptions and meaning.

Treatment consisted of seventy-five hours of time divided in into fifty minute class periods five days per week for 18 weeks. Students learned new programming concepts an average of two days per week. These topics were typical for an introductory programming course and included variables, conditionals, loops, subroutines, arrays, computer graphics, and an introduction to object-oriented concepts. An average of three days per week was spent on lab activities where students were asked to apply and extend the concepts taught in the course. Each programming assignment was designed to stretch the student to make new connections on their own with the concepts as opposed to simple "proof of concept" labs. The goal of this was to encourage mastery learning rather than memorization. For example, one lab assignment that was given after a lesson on conditional statements required students to use a nested conditional to meet one of the specifications. This forced to students to deduce the possibility of nesting and augmented their understanding of the concept.

5. Results

Scores on the CCTT-Z are computed based on the number of correct responses. The test consists of 52 questions yielding a maximum composite score of 52. This score can be further subdivided into the specific sub-areas of reasoning. Below, results are presented for the entire study population followed by results of the accelerated and honors population in isolation. For each of these groups, both the composite and sub-scores are examined. See Table 2 for a summary of mean difference, Table 1 for a summary of T-Test results on the mean differences, and Table 3 for a summary of standard deviations in mean scores.

5.1 Entire population

Beginning with the entire sample population, the experimental group had a mean composite score of 29.17 (*SD* = 3.66) compared to a mean score of 28.00 (*SD* = 6.02) for the control group. However, the difference in the mean composite scores was not statistically significant ($t = 0.89$, $df = 47$, $p > 0.05$, one-tailed test). For the deductive reasoning sub-score covered by 24 of the 52 assessment questions, the mean of the experimental group was 14.17 (*SD* = 2.78) compared to a mean deductive score of 13.33 (*SD* = 3.22) for the control group. Again, this difference was not statistically significant ($t = 1.07$, $df = 58$, $p > 0.05$, one-tailed test).

For the inductive reasoning sub-score covered by 18 of the 52 assessment questions, the mean of the experimental group was 10.83 (*SD* = 1.68), which was identical to the mean control group score of 10.83 (*SD* = 2.97). The inductive sub-score means showed no statistical significance ($t = 0$, $df = 46$, $p > 0.05$, one-tailed test).

For the credibility and observational reasoning sub-score covered by 4 of the 52 assessment questions, the mean of the experimental group was 2.13 (*SD* = 1.07) compared to a mean score of 1.67 (*SD* = 0.99) for the control group. The difference in the means was statistically significant (t = 1.75, df = 58, $p < 0.05$, one-tailed test).

For the assumption reasoning sub-score covered by 10 of the 52 assessment questions, the mean of the experimental group was 5.67 (*SD* = 1.60) compared to a mean assumption score of 5.43 (*SD* = 1.77) for the control group. This difference was not statistically significant (t = 0.53, df = 58, $p > 0.05$, one-tailed test).

Table 1: T-Test Results for Mean Differences in Composite and Sub-Scores.

Group	**t-value**	**df**	**p**
All			
Composite Score	0.89	47	0.19
Deduction	1.07	58	0.14
Induction	0.00	46	0.50
Credibility and Observation	0.72	58	0.04
Assumptions	0.53	58	0.30
Meaning	-0.31	58	0.34
Accelerated Only			
Composite Score	1.29	26	0.10
Deduction	1.70	26	0.05
Induction	0.08	26	0.47
Credibility and Observation	1.24	26	0.11
Assumptions	0.60	26	0.28
Meaning	-0.41	26	0.34
Honors Only			
Composite Score	-0.80	24	0.21
Deduction	-0.07	24	0.47
Induction	-2.09	24	0.03
Credibility and Observation	0.72	24	0.24
Assumptions	0.25	24	0.40
Meaning	-0.93	24	0.18

Finally, for the meaning reasoning sub-score covered by 15 of the 52 assessment questions, the mean of the experimental group was 6.77 (*SD* = 1.77) compared to a mean score of 6.97 (*SD* = 1.97) for the control group. This difference was not statistically significant (t = -0.41, df = 58, $p > 0.05$, one-tailed test).

5.2 Accelerated participants

Looking at the results for the fourteen accelerated (middle academic level) participants only, the mean composite score for the experimental group was 28.07 (*SD* = 3.79) compared to a mean composite score of 26.14 (*SD* = 4.09) for the control group. This difference was not statistically significant (t = 1.29, df = 26, $p > 0.05$, one-tailed test).

For the deductive reasoning sub-score, the mean of the experimental group was 13.50 (*SD* = 2.93) compared to a mean deductive score of 11.71 (*SD* = 2.61) for the control group. This difference was very close to statistically significant (t = 1.70, df = 26, p = 0.05, one-tailed test).

For the inductive reasoning sub-score, the mean of the experimental group was 10.71 (*SD* = 2.33) compared to a mean inductive score of 10.64 (*SD* = 2.44) for the control group. This difference was not statistically significant (t = 0.08, df = 26, $p > 0.05$, one-tailed test).

Table 2: Mean Composite and Sub-Scores on the CCTT-Z for Research Participants.

Group	**C**	**I**	**D**	**Cr**	**A**	**M**
All						
Experimental	29.17	10.83	14.17	2.13	5.67	6.77
Control	28.00	10.83	13.33	1.67	5.43	6.97
Accelerated Only						
Experimental	28.07	10.71	13.50	2.00	5.00	6.21
Control	26.14	10.64	11.71	1.50	4.64	6.79
Honors Only						
Experimental	30.54	10.77	15.38	2.31	6.46	7.15
Control	32.00	12.08	15.46	2.00	6.31	7.85

C = composite score; D=deductive; I = inductive; Cr = credibility and observation; A = assumptions; M = meaning

Table 3: Standard Deviations of Composite and Sub-Scores on the CCTT-Z for Research Participants.

Group	**C**	**I**	**D**	**Cr**	**A**	**M**
All						
Experimental	3.66	1.68	2.78	1.07	1.60	1.77
Control	6.20	2.97	3.22	0.99	1.77	1.97
Accelerated Only						
Experimental	3.79	2.33	2.93	1.24	1.57	1.63
Control	4.09	2.44	2.61	0.85	1.60	1.37
Honors Only						
Experimental	3.48	0.83	2.22	1.03	1.45	1.91
Control	5.58	2.10	2.96	1.15	1.70	1.91

C = composite score; D=deductive; I = inductive; Cr = credibility and observation; A = assumptions; M = meaning

For the credibility and observation reasoning sub-score, the mean of the experimental group was 2.00 (*SD* = 1.24) compared to a mean credibility score of 1.50 (*SD* = 0.85) for the control group. The difference in the means was not statistically significant (t = 1.24, df = 26, $p > 0.05$, one-tailed test). For the assumption reasoning sub-score, the mean of the experimental group was 5.00 (*SD* = 1.57) compared to a mean assumption score of 4.64 (*SD* = 1.60) for the control group. This difference was not statistically significant (t = 0.60, df = 26, $p > 0.05$, one-tailed test). Finally, for the meaning reasoning sub-score, the mean of the experimental group was 6.21 (*SD* = 1.63) compared to a mean score of 6.78 (*SD* = 1.67) for the control group. This difference was not statistically significant (t = -1.01, df = 26, $p > 0.05$, one-tailed test).

5.3 Honors participants

Looking at the results for the thirteen honors (highest academic level) participants only, the mean composite score for the experimental group was 30.54 (*SD* = 3.48) compared to a mean composite score of 32.00 (*SD* = 5.58) for the control group. This difference was not statistically significant (t = -0.80, df = 24, $p > 0.05$, one-tailed test).

For the deductive reasoning sub-score, the mean of the experimental group was 15.38 (*SD* = 2.22) compared to a mean deductive score of 15.46 (*SD* = 2.96) for the control group. This difference was not statistically significant (t = -0.07, df = 24, $p > 0.05$, one-tailed test).

For the inductive reasoning sub-score, the mean of the experimental group was 10.77 (*SD* = 0.83) compared to a mean inductive score of 12.08 (*SD* = 2.10) for the control group. The difference in the means was statistically significant (t = -2.09, df = 16, $p < 0.05$, one-tailed test).

For the credibility and observation reasoning sub-score, the mean of the experimental group was 2.31 (*SD* = 1.03) compared to a mean credibility score of 2.00 (*SD* = 1.15) for the control group. The difference in the means was not statistically significant (t = 0.72, df = 24, $p > 0.05$, one-tailed test).

For the assumption reasoning sub-score, the mean of the experimental group was 6.46 (*SD* = 1.45) compared to a mean assumption score of 6.31 (*SD* = 1.70) for the control group. This difference was not statistically significant (t = 0.25, df = 24, $p > 0.05$, one-tailed test).

Finally, for the meaning reasoning sub-score, the mean of the experimental group was 7.15 (*SD* = 1.91) compared to a mean score of 7.85 (*SD* = 1.91) for the control group. This difference was not statistically significant (t = -0.93, df = 24, $p > 0.05$, one-tailed test).

5.4 Summary of significant differences

Comparing the experimental results against the research hypothesis that students treated with computer programming will score higher on a standardized reasoning test has yielded mixed results. Mean scores on the assessment tool were, in general, higher for the experimental group than the control group. These higher means included the composite score along with four of the five sub-scores (induction, deduction, credibility and observation, and assumptions). These higher means were also seen when looking at the results of the accelerated-level students in isolation. Most of these mean differences were not statistically significant, but the fact that thirteen of the eighteen computed means, and ten out of twelve means if the honors-level participant results are omitted, were higher for the experimental group lends some support to the research hypothesis.

Further supporting the research hypothesis, one of the mean differences was shown to be statistically significant, and one was shown to be on the border of significant. For specific numbers, see Table 4 which summarizes the statistically significant findings. Looking at the entire sample, the higher mean for the experimental group in credibility and observational reasoning was shown to be significant. It should be noted, however, that results computed from this and other sub-scores must be taken with caution given that they are composed of a smaller number of questions. The credibility and observational reasoning sub-score in particular was assessed using four questions. That said, these results are still interesting and surprising since credibility and observation might not automatically be associated with computer programming activities.

Table 4: Summary of Statistically Significant T-Test Results for Mean Differences in Composite and Sub-Scores.

Group	**t-value**	**df**	**p**
All			
Credibility and Observation	0.72	58	0.04
Accelerated Only			
Deduction	1.70	26	0.05
Honors Only			
Induction	-2.09	24	0.03

Looking at only the accelerated-level students, the higher mean deductive reasoning sub-score was shown to be on the border of significance with a p-value of exactly 0.05. This result suggests that computer programming may have a positive impact on deductive reasoning skills. This would make sense considering that computer programming activities require deduction-like thought processes where participants systematically put ideas together to form a solution to a problem. A larger sample may very well confirm that this result is in fact significant.

Another interesting result came from examining only honors-level students. This experimental sub-group demonstrated no benefit from the programming treatment.

In fact the control had a higher mean composite score and higher means in all sub-scores except credibility/observation, and reasoning about assumptions. The difference in the inductive reasoning sub-score means (higher for the honors-level control group) was even statistically significant (See Table 4). One possible explanation for this is the fact that honors-level students have already developed their reasoning skills through either natural ability, more difficult course work, or both. Therefore the curriculum used in the treatment may not be adequately challenging to further boost their reasoning abilities.

6. Conclusions and future work

The goal of this study was to discover a link between learning computer programming and enhanced reasoning skills in high school students. Based on the summary of results from the previous section, the research hypothesis cannot be confirmed, but it cannot be rejected either. There is some compelling evidence to suggest that learning computer programming may have cognitive benefits in terms of reasoning, and further research needs to be done.

While the results were unable to confirm this connection they were certainly compelling enough to warrant further study. Suggestions for future studies would include replacing the most gifted students from the study with more standard or accelerated students. The results of this study clearly demonstrated that the most gifted students in this sample did not benefit from the computer programming treatment, while the evidence for the rest of the students was more compelling. Additionally, the authors are planning on a much larger study involving roughly one hundred students and implementing a pre-test, post-test methodology to better determine the potential improvements for each individual student as well as overall population improvements as was provided in this study. The potential issue with this change, however, would be confusing natural cognitive development with cognitive development caused by the treatment.

In terms of suggestions for the measurement tool, choosing one that provides more items in specific reasoning sub-areas would allow better conclusions to be drawn about these areas. In particular, an assessment tool that focuses on credibility and observational reasoning would be an interesting follow-up to this study since a significant difference was noted in the experimental group.

In terms of suggestions for enhancing treatment in future studies, one possibility would be to supplement the programming curriculum with transfer training that helps students relate the reasoning concepts in programming with their life applications. At least one study concluded that this sort of treatment modification yielded positive results [10]. Of course this sort of curricular modification raises questions as to the real cause of any observed benefits. Likewise, a treatment that emphasizes a lab-driven, problem-solving oriented curriculum will likely yield better results than a lecture- and demonstration-based treatment. This approach follows the philosophy that reasoning is best improved through practice. Finally, the instructional strategies used in teaching computer programming must focus on mastery learning because a student who does not fully grasp the programming concepts is unlikely to reap cognitive benefits. This notion is supported by a longitudinal study that found students who had only experience as opposed to expertise were unable to transfer those skills to other areas [6] [11].

By implementing one or more of the suggestions above it is very possible that some sort of link could be discovered. Ultimately identifying a connection between reasoning skills and learning computer programming has two potential key benefits. First, it will help make the case for its inclusion in high school and college curricula as not only a skill, but as another tool to enhance cognitive skills. Second, identifying this connection could help to shape the future of computer science education by revealing new and better pedagogical strategies for teaching and learning computer programming.

7. References

[1] Burton, J. K. and S. Magliaro, "Computer programming and generalized problem-solving skills: In search of direction", *Computers in the Schools*, 4(3/4), pp. 63-90, 1987/88.

[2] Clements, D., "Effects of LOGO and CAI environments on cognition and creativity", *Journal of Educational Psychology*, 82(1), pp. 309-318, 1986.

[3] Clements, D. and D. Gullo, "Effects of computer programming on young children's cognition", *Journal of Educational Psychology*, 76(6), pp. 1051-1058, 1984.

[4] Ennis, R., J. Millman, and T. Tomko, *Cornell Critical Thinking Tests: Manual, test* (5th Ed). Seaside, CA: The Critical Thinking Company, 2005.

[5] Leher, R., T. Guckenberg, and O. Lee, "Comparative study of the cognitive consequences of inquiry-based LOGO instruction", *Journal of Educational Psychology*, 80(4), pp. 543-553, 1988.

[6] Kurland, D. M., et al. "A study of the development of programming ability and thinking skills in high school students", *Journal of Educational Computing Research*, 2(4), pp. 429-458, 1986.

[7] Mayer, R. E., & Fay, A. L., A chain of cognitive changes with learning to program in LOGO. *Journal of Educational Psychology*, 79(3), 269-279, 1987.
[8] Miller, R. B., G. N. Kelly, and G. T. Kelly, “Effects of Logo computer programming experience on problem solving and spatial reasoning”, *Contemporary Educational Psychology*, 13(4), pp. 348-357, 1988.
[9] Palumbo, D. B., “The effect of BASIC programming language instruction on high school students’ problem solving ability and computer anxiety”, *Journal of Research on Computing in Education*, 23(3), pp. 343-372, 1991.
[10] Unuakhalu, M. F., “Enhancing problem-solving capabilities using object-oriented programming language”, *Journal of Educational Technology Systems*, 37(2), pp. 121-137, 2009.
[11] VanLengen, C. A. and C. D. Maddux, “Does instruction in computer programming improve problem solving ability?”, *Journal of Information Systems Education*, 2(2), pp. 11-16, 1990.
[12] J. Holvikivi, “Logical Reasoning Ability in Engineering Students: A Case Study, *IEEE Transactions on Education*, vol. 50, no. 4, pp. 367-372, 2007

A Microcosm for Effective Undergraduate Education of Computer Science

Andrzej Bieszczad
California State University Channel Islands, One University Drive, Camarillo, CA 93012, USA

Abstract - *In this paper, we describe the microcosm created at California State University Channel Island (CSUCI) for effective education of Computer Science and Information Technology. We claim that our learner-centered environment is critical in effective education of our student population. We describe how our multifaceted and holistic approach to teaching hard subjects results in a substantial gain in students knowledge and skills that allow most of our graduates to be hired on the spot; often even before completing the graduation. We believe that the substantial value added by our program as measured by the increase in knowledge and skills at the time of entry (enrollment) and the time of exit (graduation) makes the Computer Science Program at CSU Channel Islands a very attractive destination for candidates looking for a top-notch technical education.*

Keywords: computer science, education, quality, learning, hands-on, experiential.

1 CS@CSUCI Overview

California State University Channel Islands is a relatively new campus which since 2002 has been trying to provide quality higher education to the population of Ventura County. It is the only public university in the county; the county that over the recent years has seen considerable growth with the population and industry migrating from the more populous and expensive Los Angeles county that includes nearby San Fernando Valley, and the cities in the Los Angeles basin.

The university has tried to grow in times doomed by cuts to the public spending and a dramatic decline in the annual budgets of its mother California State University system.

1.1 Programs

The Computer Science Program at CSUCI offers six programs: Master of Science in Computer Science (MSCS), Bachelor of Science in Computer Science (BSCS), Bachelor of Science in Information Technology (BSIT), Minor in Computer Science, Minor in Game Design and Development, and Minor in Robotics Engineering that the Academic Senate of CSUCI has just voted to offer in Fall 2012. All programs follow the guidelines published by the Association for Computing Machinery (ACM [1]) and by the Accreditation Board for Engineering and Technology (ABET [2]). The program has not formally applied for accreditation due to the lack of sufficient resources. Instead, all efforts are put into the direct education of our students.

1.1.1 Bachelor of Science in Computer Science

As shown in Figure 1, the Computer Science curriculum is a strict program that has some unique components that we will discuss in the following sections. There is a considerable collection of Math courses that teach the students a disciplined approach to problem solving. Many of our students are in fact double majors with Mathematics being the second major.

1.1.2 Bachelor of Science in Information Technology

The BSIT program stresses the training in system, network, and database administration. It includes courses that will allows the students to be also Web masters and Web programmers. These professionals usually do not require the same level of Mathematics education as Computer Science graduates.

1.1.3 Minor in Computer Science

The Minor in CS is an attractive proposition to a variety of students from other majors that want - or need - to acquire basic understanding of computers and programming skills. Many students from Mathematics, Business, and Arts take upon that opportunity, but at any moment we have a number of students from other programs as well: Psychology, Biology, Sociology, and so on.

1.1.4 Minor in Robotics Engineering

This new program will target mostly students from STEM disciplines by providing a collection of course offerings that allows the graduates to pursue careers in a variety of fields that employ all sorts of robots. The courses provide theoretical foundations from Physics and Mathematics, along with practical skills in electronics, hardware, embedded systems, and application software; especially in Artificial Intelligence and Image Processing.

1.1.5 Minor in Game Design and Development

The Minor in Game Design and Development usually attracts students from Arts and Computer Science. It is a multidisciplinary program that caters to the students, who want to work in the entertainment industry as art designers, programmers, project managers, etc. The program's focus is on the integrative aspects of game design; it leaves the specific skills (such as art design, programming, project management, etc.) to the major programs.

1.1.6 Master of Science in Computer Science

The graduate program offers a formal thesis-based education that requires substantial research effort. The program is

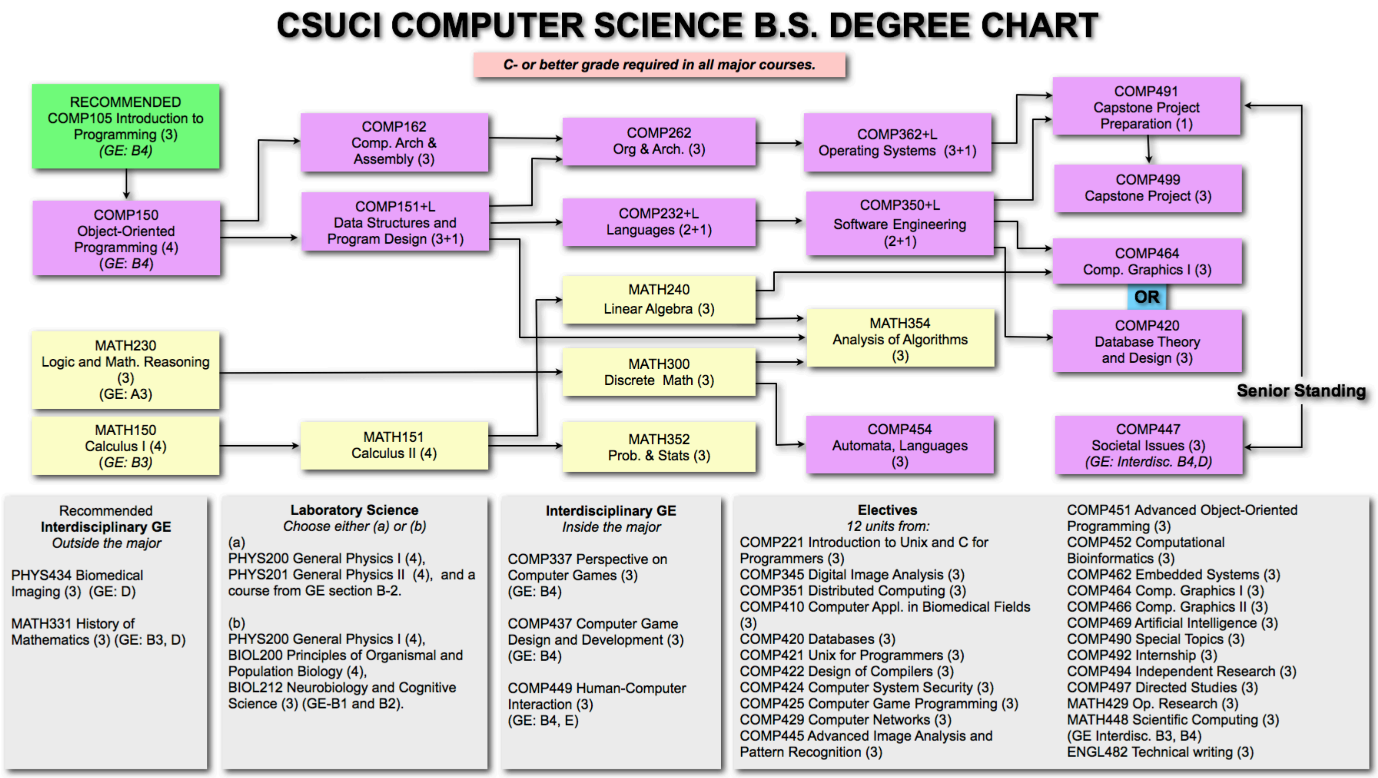


Figure 1: Diagram showing the Bachelor of Science in Computer Science curriculum.

managed financially by the CSUCI's Extended University. It has succeeded in proving that a self-support at funds-strapped public institutions is a viable option if the students receive good return for their investment.

2 Challenges

Let's examine the challenges that a computer science program - and faculty - face at CSU Channel Islands. We believe that many of the items on the following laundry list apply to the majority of public, primarily teaching, undergraduate universities in the Nation. Some of the challenges are overexposed at CSUCI, because the university has not managed to gather a momentum in size and funding that may alleviate to some extent some of the problems.

2.1 Tough Times for Education

Educating a Computer Science professional is a very complex task. It takes plenty of resources both human and technological to equip graduates with the skills that will allow them to find - and keep - a job and - if she or he chooses so - to pursue further studies at a graduate level. The task is especially challenging at under-funded public primarily undergraduate universities that are not able to cover the budget gaps with external funding. California State University system falls in that category, and as of the time of writing (March/April 2011), it seems that things will get much worse before getting any better. Due to many factors, the fiscal axe affects equally all units of the university; that puts the STEM programs in especially difficult position, as they require numerous lab-centered activities that are resource-intense, and therefore very costly. The Computer Science program is not an exception.

2.2 Patching Up High School Education

The universities - especially the public genre - have been extending their General Education components over years. In large part that is the consequence of the failing middle-level education in the U.S., whose high schools are not fulfilling their roles and wasting vast amounts of public funding in graduating students that very often need remedial courses to patch up the basic Math and English skills. While providing well-rounded education is a noble goal (that - by the way - should be mostly done in a high school), the growth of the General Education components of the curriculum and minimizing the time-to-graduate by capping the number of units in the program is another serious challenge to educating professionals, who not only need to acquire substantial knowledge in Mathematics, Science, and Computer Science (see Figure 1 for an example), but also need practical skills - like programming, testing, and debugging - that take very long time and effort to acquire and master. Computer Science is a constructive field - like Engineering - and becoming proficient in a particular algorithm requires practicing its application in a variety of contexts; a substantial time commitment.

2.3 STEM Crisis

CSUCI is one of the smallest campuses with relatively small STEM programs and no engineering school. The Computer Science program needs to compete for means with other STEM disciplines that even when put together are overshadowed by Humanities, Business, and Arts. Only Biology (that received substantial support from the local industry giant, Amgen) can be compared size-wise with the largest programs. Computer Science with its five state-funded programs is a medium size program with approximately 150 FTES (Full-Time Equivalent Student; a CSU measure of program size that determines the level of funding).

The graduate program is self-supporting, and does not draw from the state funds; as we will see momentarily, quite opposite is true.

2.4 Is Computer Science an Experimental Science?

Quite often the needs of the Computer Science programs are perceived as not as demanding as the classical Science programs such as Biology, Chemistry, or Physics, especially in schools that do not have an engineering school. For example, lab fees collected from students in addition to the university fees can be used to cover many expenses of a Biology lab. Unfortunately, it is virtually impossible to equip a Computer Science lab using the same fees due to the restrictions on the spending that years ago were geared towards classical science labs, and have not evolved to adapt to the new reality that requires buying things like software.

2.5 Faculty

The increasing reliance on contingent faculty is a severe problem for small programs, since the scarce tenure and tenure-track resources have to commit large portion of their time to indirect activities (management, service, etc.). With the improving job market, attracting highly educated holders of Ph.D. degrees to low-paying (by the technology industry standards) non-tenure, temporary faculty jobs is very difficult. Unfortunately, the egalitarian atmosphere at academia prevents effective addressing of that issue, as the share of the fiscal burden is distributed evenly.

2.6 Student Body

Public higher education in California is regulated by the Master Plan for Higher Education of 1960 ([3]) that created a three-tier system. At the top, there is University of California (UC) that is a research-oriented, doctorate-granting system of ten campuses that include such gems as UC Berkeley and UCLA. Every Californian in the top 12.5% of the high school class has the right to a place in the system. Additionally, the system reserves spots for best-performing students from the two other tiers if they choose to pursue studies at UC. The second tier, California State University (CSU), is primarily an undergraduate teaching institution that only recently gained a right to grant doctorate degrees in Education and Physical Therapy. Many campuses have graduate schools, but they commonly provide professional training (Master's or credentials). Every student in the top 33% of a high school class has a place at CSU that is guaranteed by the law. Community Colleges constitute the lowest tier of the postsecondary education in California. While in the large part they provide a cheap access to education to the most underprivileged parts of the society, they are also commonly selected as an inexpensive way of taking the General Education requirements out of the way, before focusing on the major core courses at a higher-tier university.

As a result of the legislation, few much-less-funded CSU campuses can compete for top students with the UC (and other top universities in California such as Stanford, CalTech, USC, or Harvey Mudd College) - CalPoly being a primary example in engineering fields. Although at CSUCI we hope to match their prestige in the future, in reality, at the moment many of the students coming to CSUCI are older, returning students, or those that under-performed in one way or another, and therefore did not qualify for a spot at a higher-rated institution. This is not to say that all bright students shy away from CSUCI, because that would be far from the truth. Quite to the contrary, there are many excellent students coming to our university; nevertheless, it's hard to deny that many of our students struggled with all sorts of problems along the way. In the past, we saw students with serious family problems, severe cases of "senioritis", lack of interest in peripheral subjects, introversion common in students gifted in STEM fields, and personality issues ranging from mild ADHD to serious cases of autism.

3 Microcosm

Seeing a litany of challenges listed in the previous section one may conclude that teaching Computer Science in such environment is close to impossible. Fortunately, that is not so. We are proving that we are able to educate professionals that do succeed in their careers in spite of all the obstacles. However, an exceptional effort is required from the faculty not only in the classroom, but also outside and at the curriculum design board.

Fortunately for the Computer Science program at CSUCI, one of its major problems - the young age - is also its blessing that allows for a relatively fast adoption of new teaching methods and tools, and fairly painless adaptation to the evolving curriculum. Implementing some of the elements of the microcosm that we created for our students at CSUCI could have faced much more resistance under different circumstances. We can clearly see that in this regard it pays off to be a startup program.

3.1 Setting Standards High From the Onset

One of the most important decisions early in the evolution of the program was the stress on the quality. From the onset, we established an atmosphere that tells the students that while we do not have a reputation of Caltech or Berkeley, we provide them with education that not only matches, but in many respects surpasses more recognizable institutions. They quickly realize that the effort will have to match the expectations that are set very high. That realization enables in many students self-confidence and belief in the ultimate success. We are convincing them that if they put an effort, they will be the winners denying the odds. We actually publish a formal document "Expectations for Computer Science Students" that is distributed every semester to all students and posted on the Web and in the labs.

3.2 Solid-Rock Curriculum

A rock-solid curriculum (like the one in Figure 1) that is infused with strict Math and Science courses is the foundation of setting the bar high. It's a bit of a double-edge sword as it scares some potential candidates, or forces some others to leave if they are not willing to work hard in spite of the program, faculty, peers, and the university doing all they can to help. Some students elect to pursue the degree in Information Technology that is lighter on the Mathematics component, and some just fall off, or switch programs. While lowering the standards might be tempting to increase the enrollment, we were in fact able to attract more capable students as the result.

3.3 Small Sections

A small program is a challenge to manage with the administration being strict on section sizes in times of budget cuts, but it also constitutes an environment in which a student does not become an anonymous mass customer; when classes are large we hardly see individuals. The other factor that enforces relatively small sections at CSUCI is the localization in old buildings that restrict the sizes of the labs and lecture halls. As a result, most of our sections are capped at 20-25 students; instructors indeed know the names of all the students in the class. That allows the meetings to be very interactive. Instructors use the instant feedback to customize the sessions to accommodate the needs specific to a given cohort of students that naturally change from year to year.

3.4 Direct Access to Instructors

Most of the Computer Science sections are taught by faculty instructors; that includes the labs. Since the sections are small, students get their questions addressed by those, who are best qualified to answer them. We engage student tutors, but they work under the close supervision of the faculty in a helper capacity.

3.5 Inter-Course Mobility

Another advantage of a program with relatively little red tape and a direct day-to-day evaluation by the faculty is student mobility between courses. Some students coming to our program as transfers from colleges, or with credits from advanced placement courses in high schools, cannot cope with the requirements of courses in which they are entitled to enroll. We meticulously advise these students that they should take alternate or pre-requisite courses even if there is no formal requirement. While some students are not comfortable with such advice, many expressed gratitude in the following semesters; sometimes even claiming that the extra step triggered a complete different flow of events.

3.6 Inter-Program Mobility

While we always hope that restructuring curricula to match individual needs will address the emerging issues, sometimes the gaps in knowledge or skills are just too large to fix. In such cases, we persuade affected students to follow an alternate course of study in the Information Technology program, or, we help students to transfer to an alternate field of study. The university is very accommodative in that respect, as we want every student to succeed ultimately. Unfortunately, we had numerous cases of students misguided by high school counsellors in selection of their future careers. Few incoming students realize how strict the Computer Science program is; many can adapt, but for some the sacrifice - often the lack of the broadly-understood social life - is hard to accept.

3.7 Peer- and Team-Based Curriculum

Since the program has been offered for the first time in 2003, the courses have evolved from mainly lecture-based to complex hybrids that involve lectures, seminars, and labs. While certain objectives of the curriculum are achieved through individually-administered homework, others are attained through teamwork. That may take a form of project teams (e.g., in Software Engineering) or peer programming (e.g., Operating Systems). Teamwork is especially effective when the team membership is rotated, as it leads to opening of students to the whole class community, and results in dissemination of knowledge and skills through the network of peers.

3.8 Hands-on Laboratories

It is hard to imagine good Computer Science education without substantial time in laboratories. The original curriculum at CSUCI did not include formal labs, in part due to the perception of the program being closer to Mathematics rather than experimental Science. While some students can work at a theoretical level and laboratory experiments independently, due to a variety of reasons the majority of our students were not able to make appropriate time commitment without formal labs. Therefore, we have added labs over the last few years and have seen great improvement in student learning outcomes as the result (see Figure 2 and Figure 3 later in the text).

3.9 Strict Deadlines

Quite often computer professionals need to work under a stress of approaching deadlines. It is a duty of an educational institutions to enforce strict deadlines, so the graduates are not in shock when they join "the real world". There is however an important systematic aspect of enforcing deadlines; meeting them ensures that the students actually build up their knowledge and skills incrementally. Like in many formal fields, missing a pillar here or there destroys the whole construction. The Computer Science program at CSUCI is a power user of learning management systems (Blackboard [4]) that provides convenience of applying the enforcements with little effort.

3.10 Capstone

The span of one course, even with an extended lab component, is too short to allow a student to build a portfolio that can showcase her or his abilities. Therefore, we put in place a two-stage capstone project course that gives students a vehicle to do just that. The first part is a one-unit capstone preparation course in which students work with faculty advisors on finding a topic for a capstone project and creating its functional specifications. The 3-unit second part of the course in the following semester is the actual implementation of the design.

Splitting the project into two parts had a very positive impact on the outcome of the effort; prior to that, the students would often spend a large portion of the semester trying to pinpoint the focal point. Currently, they are ready to work from the start of the semester, leaving time at the end of the semester for writing a project report that is a mandatory component of the course.

3.11 Self-Supporting Graduate Program

While the graduate program does not have a direct impact on the undergraduate education, it has been a blessing for the Computer Science program. The Extended University redirects through Continuing Education Revenue Fund (CERF) parts of the income from the self-supporting programs to the state-side academic units that offer the programs. The funds are not huge but allow the Computer Science program to budget many necessities that in times of short budgets sound like luxuries.

Owing to the CERF program, the Computer Science program at CSUCI has been infused annually with so much needed means from the self-supporting graduate program. Most of the equipment in specialized labs as well as adapting space to the lab needs are covered by CERF. Some special projects like Robot Soccer undergraduate research were possible only because of the tunneling funds from the Extended University.

3.12 Tutoring

Then, there is student tutoring; one of our greatest successes. Using the CERF, we were able to adopt some space surrounding the labs and create peer tutoring environment. Every semester, we pick the brightest students and offer them tutoring jobs paid partially by CERF and by student lab fees. The tutoring center has been a huge success with dozens of students visiting the center daily looking for assistance with homework, labs, projects, etc. We educate our undergraduate tutors that the help that they provide needs to be in line with the university code of ethics. We have not had any problems with tutors crossing the line, since the tutors are hand-picked and they do know that their very desirable job is in stake.

We also use the tutors in class to help with labs in a supplemental role. Some of the labs are intense and quite often even with a small section one instructor is not able to handle all the traffic.

The tutors also play an important role in educating students with special needs; for example, with ADHD and autism. The university has a Disability Assistance Services (DAS), but there are no provisions for assisting special students in classroom. The tutors help instructors with high-maintenance students, so the impact on the reminder of the class is manageable.

While the student tutors are paid for the work, many has expressed gratitude not so much for the money as for the opportunity to teach. As Stephen Braken said: "The best way to learn is to teach..." ([5]).

In a recent anonymous survey of the tutoring, numerous Computer Science students evaluated the services very highly.

3.13 Undergraduate Research Fair

Most of the CSUCI students learn best in concrete, hands-on experiential sessions. The university put in place a number of programs that help with funding such a renaissance-man approach to education; e.g., faculty mini-grants and research courses. The Computer Science program was successful on several occasions in securing such grants. Together with the funding coming from CERF, the students were involved in Robot Soccer and Neural Forecaster projects. Some of that efforts resulted in conference publications.

Every year, the university celebrates faculty-student research through a formal fair in which students presents their research in oral or poster presentations. Through the participation in the fair, students experience a conference-like settings. Traditionally, Computer Science students presence in the event is abundant.

3.14 Extra-Curricular Activities

CSUCI students has been competing in the ACM programming contests since the first year. In fact, CSUCI is one of the largest groups participating in the South-West Region competition at Riverside Community College. The event mobilizes the students to extra studies, but most of all build a peer network that subsequently projects positively on all aspects of the Computer Science program.

While the Computer Club formally falls into a Student Affairs jurisdiction, the faculty was very instrumental in creating the club and setting it on a successful track. Currently, the club has many members and is very active on several fronts: from game development seminars, through technical presentations and programming contests, to game nights.

The event that cemented the first group of computer club members was an international trip to Europe to visit sister programs at other universities. The trip was organized by the faculty and funded by the Instructionally Related Activities (IRA) fund.

3.15 Learning Management System

A tool is useless without a proper use, so we do not claim that the use of Blackboard guarantees educational success. However, from our experience it is evident that proper use of the tool facilitates learning and overall is a very positive factor. We need to note that our faculty uses the Blackboard at an expert level committing substantial time to setting up and maintaining of the courses. Without that commitment, Blackboard could have appeared as a nuisance to both the students and the faculty.

3.16 Cafe at the Labs

We utilized the room around the Computer Science labs to create an informal retreat cafe. There are only a few vending machines there, but the place has several tables, and some of the faculty even brought a few leather couches, so the students can relax between the classes. The space has become a popular hanging place. We utilize the walls for posting formal and informal messages about the program. The space facilitates interactions between the students and allows them to seek help in informal settings.

4 Experiences in Teaching Foundations of Operating Systems

This author piloted many of the innovations in several courses; most recently in the undergraduate Operating Systems course. The course used to be an old-fashion lecture-type 3-unit lecture with theoretical homework and some in-class exercises that needed completion at home. Since Fall 2011, the course has followed the 3+3 formula; that is, there are three hours of lecture and 3 units of lab. In fact, there is just one 1.5-unit classical - albeit quite interactive in the small settings - theoretical lecture that is followed by a homework. The homework has a hard deadline before the next segment of a weekly schedule; that is the second 1.5-unit of a lab-focused informal lecture that explains to the students techniques, methods, libraries, utilities, etc., that they will use in the labs. It usually includes small exercises that allow the students to test fundamental concepts studied in that particular week. The second part extends to the next segment, the 3-unit lab that involves a substantial multi-stage project that the students usually need to finish at home.

The students work individually on their homework, but the labs are peer-based. Peers rotate every week, so all students have a chance working with most of the others. That is an important element of peer-based activities, since otherwise, in a fixed peer group, the roles of alpha and beta collaborators are assigned permanently. That leads to uneven efforts, frustration, and unbalanced learning in the end. With rotating peers, the teams are freshly set on a weekly basis. It has been our observation that the workload is much better balanced in that way. To ensure that there is absolutely no free rides, the peers report work distribution quantitively (using percentages) as part of their mandatory lab commentary, and the reports are used to weight contributions appropriately. The students have been quite honest in their reports.

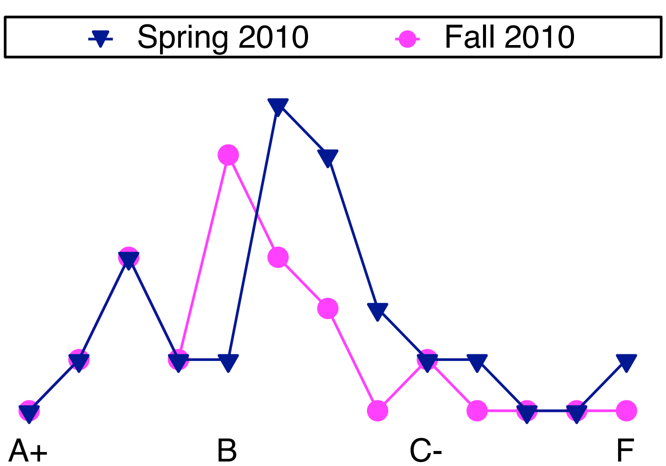


Figure 2. Improvement in student grades seen especially in the middle of the range.

The deadlines for lab submissions are also strictly enforced through the visibility feature in Blackboard. Simply, homework and the labs appear for the submission session, and automatically disappear when the deadline passes. They appear again after they have been graded. Each students get (or a peer-group for labs) receives individualized feedback, but the results are not published.

All other course management is done through Blackboard as well. Students have access to all course information including a syllabus with clearly stated student outcomes; all lecture notes that also appear weekly when a new topic is announced; homework and labs; gradebook; discussion boards; and last but not the least, the calendar. Blackboard has many problems with proper calendar scheduling, so maintaining the course calendar is a substantial effort, however it's very worthwhile as student feedback emphasizes the importance of the keeping the calendar up to date. The gradebook could be better, but it does the job - with some effort. Another important element enabled by the learning system is reporting to the students in a timely fashion the current overall weighted grade from the course. Some students in the past were surprised that the components of the course are not weighted in the same way.

In fact, to maintain a balance between the theoretical and practical aspects of the course, the students are graded as follows:

- 20% - for 15 homework assignments
- 30% - for 10 labs
- 20% - for 2 projects
- 30% - for the final exam.

The theory consist of the classical material based on the Silberschatz's textbook ([6]). The lab part is a programming-intense journey into experimenting with theoretical concept in the world of command-line Linux, wide spectrum of Unix system programming, POSIX threads, FUSE, and Linux device drivers. One of the important aspect of the course is also its emphasis on C programming, since the CSUCI students are educated in Java throughout their lower division courses.

Most of the material-related offline questions are handled through discussion boards. That optimizes the time that the instructor must commit to handle student questions, since many students face similar problems and ask the same questions. All students subscribe to the feed in the discussion board, so they get notifications if there is any activity.

One of the traditional problems with educating students in system-level skills has been access to protected facilities on lab computers put in place by IT departments to prevent system and network abuse. We have solved that problem by running Linux (Ubuntu; [7]) in virtual machines (Oracle's VirtualBox; [8]) that the students can configure and modify to their liking.

Since this author took over the course several years ago and applied many of the aspect of the microcosm described in the previous section, there have been dramatic improvements in mastering of the course material and most of all in practical skills. That is has evidenced to some extent in the grades (Figure 2), but the most satisfying feedback is from the student evaluations in which many praise the profound effects of the course on their knowledge and self-confidence (Figure 3). To those , who are not used to the intensity in the educational approach and who sometimes complain quite bitterly, one student explained on an online anonymous forum: "this teacher expects you to really know everything what he teaches". Somehow, the academic inflation in the U.S. has made that not so obvious.

5 Can the Microcosm Survive Tough Times?

Owing to the success of many elements of the microcosm, other foundation courses are gradually adapting it's elements. After several years of working in other areas, this author will be applying the expertise to the change in teaching programming languages that have just recently obtained the extended lab component. While in the past the author applied peer-programming principles in this course, many students did not commit sufficient time to the work outside of the classroom. It will be exciting to revisit it and observe the impact of that, and the other elements of the microcosm, on the effectiveness of teaching in another foundation course in the Computer Science curriculum.

Substantial instructor effort is a critical factor in successful implementation of the microcosm. It is sad to say that with the decreasing budgets, increasingly prevalent and overloaded contingent faculty will not be able to commit sufficient time without substantial assistance (for example, from teaching assistants). To see that clearly, let's assume that that an instructor teaches 5 course section (a required course load for a contingent faculty at CSU), 20 students each, and every week, for each course, she or he grades individually a homework of 10 questions, and a lab of 4 tasks for 10 two-student peer groups in each section. If make some minimal assumptions that that to grade the instructor needs 2 minutes per each homework question and 5 minutes per a lab task, she or he needs approximately 50 hours per week (~= 5*(20*10*2+10*4*5)/60) to do just grading. Even for a tenure-track faculty with a lower, twelve-units required workload (at CSU), the time is still staggering 40 hours. Please do note that many questions take much more than 2 minutes to grade; similarly, grading a lab often is much longer than 5 minutes.

This author has found out that teaching two intense courses (Foundations of Operating Systems and Computer Graphics), advising several students in capstone project and Master research, and providing services to the university (directing the Master program, numerous curriculum revisions, 2-3 committees and task forces, working on new programs and courses, advising computer club, and managing internal and external programming contests, managing labs) was a hardly manageable workload that substantially exceeded a 40-hours work week. A prolonged exposure, or any increase in the workload, must result in sacrificing the quality of education.

If the commoditization of the Computer Science education, and STEM education in general, do not stop, the Nation will have to depend increasingly on influx of foreigners to fulfill the needs. Unfortunately, America is no longer an immigration attractor as it used to be, and therefore there is a real danger that it will loose in the not-so-distant future the

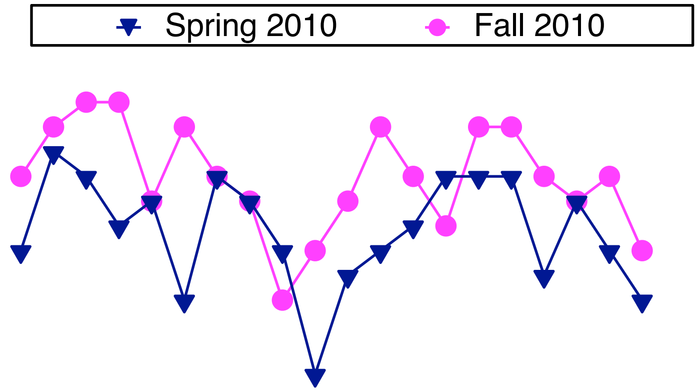


Figure 3. Increase in the evaluation of the instructor in most of the twenty categories.

technological and innovation leadership that it has enjoyed for the last century. As a nation of mere consumers, the United States will be very vulnerable to forces outside of its control; not a very attractive prospect.

6 Acknowledgments

On behalf of the CSUCI Computer Science students the author would like to express his gratitude to the administration of the university that has been accommodating the increase in the instructor costs that cover the extended laboratory components of the courses.

7 References

[1] "ACM Curricula Recommendations"; ACM, CE2008 Curriculum Update. Source: http://www.acm.org//education/curricula/ComputerScience2008.pdf.

[2] "ABET Criteria for Acrediting Computing Programs"; 2011-2012 Accreditation Cycle, Computing Accreditation Commission, ABET, October 30, 2010.

[3] "Master Plan for Higher Education". Source: http://www.cpec.ca.gov/Publications/MasterPlanIndex.asp

[4] Source: http://www.blackboard.com/

[5] Steve Bracken, "The Best Way to Learn is to Teach". Source: http://EzineArticles.com/560127

[6] Abraham Silberschatz, Peter B. Galvin and Greg Gagne , "Operating System Concepts Essentials", Wiley 2010.

[7] Source: http://www.ubuntu.com/

[8] Source: http://www.virtualbox.org/

A Study on Learning Interests of Algorithms and Data Structures

Xiaodong Wang
College of Mathematics & Computer Science
Quanzhou Normal University
Quanzhou, China
wangxiaodong@qztc.edu.cn

Jun Tian*
School of Public Health
Fujian Medical University
Fuzhou, China
tianjunfjmu@126.com

Abstract

This paper presents a linear structural equation model in analyzing the influencing factors on learning interest in the course "Algorithms and Data Structures" in computer science. Our results show that the main factors impacted on learning interest of the students are students' specialized basis and teachers' teaching strategies. Another innovation closely related to this teaching reform is a comprehensive scientific evaluation of teaching methods for this complex teaching process. This evaluation method will evaluate the whole process of teaching by a dynamic multi-factor evaluation. It will effectively promote the teaching level continues to increase.

1. Introduction

We have conducted a series of teaching reform for "Algorithms and Data Structures", the advanced course of computer science [1]. One of the most important and most difficult of the reform is to strengthen the practical aspects of the curriculum, while strengthening the teacher-student interactive teaching. The reform also brings up a problem how to accurately evaluate the effectiveness of teaching and teaching process and student's learning outcomes. The traditional approach of evaluation of teaching is through the course examination. The states of the student's mastering of the basic concepts, basic algorithms and basic techniques are evaluated by the course examination. Each student's test scores to evaluate their learning. Test scores of all students to evaluate teaching effectiveness.

A strong interest in knowledge learning can effectively enhance students' motivation for learning and mobilize the enthusiasm of students in their studies ([4], [6]). The course "Algorithms and Data Structures" is an important required course for the students majoring in Computer Science. To become qualified professional and technical personnel in computer science, it is necessary to have a good theoretical knowledge of algorithm and data structure and basic programming skills ([6]). Therefore, the study of this course is very important for students to improve their professional standards and work capability in their future academic career.

Students' learning interests have a certain relationship with environmental factors. It is an important subject of learning theory and teaching practice to analyze the influential factors in learning interests.

Interest in learning is an endogenous latent factor can not be measured directly. But it can be reported by a number of measurable variables $x_1, x_2, \ldots, x_q$.

Similarly, environmental factors to influence the students' learning interests are exogenous latent factor can neither be measured directly. But these environmental factors can also be reported by a number of measurable variables $y_1, y_2, \ldots, y_p$.

When we analyze the relationships between learning interests and their environmental influential factors, we must establish the underlying structural relationships between endogenous and exogenous latent factors.

It is not easy to analyze the relationships between learning interests and their environmental influential factors by using the traditional statistical methods.

The linear structural equation model ([3], [5])(LISREL, an acronym for linear structural relations), is an advanced structural equation modeling method. It has a path analysis of the potential factors. The method can be used to estimate the loadings of the dominate variables of exogenous latent factors $\xi_1, \xi_2, \ldots, \xi_n$ and endogenous latent factors $\eta_1, \eta_2, \ldots, \eta_m$ as the traditional method of factor analysis. It can also be used to analyze the structural relationships between endogenous and exogenous latent influential factors.

The structural relationships between endogenous and exogenous latent influential factors can also be depicted by their path graph. Therefore the method has been widely used in the field of sociology ([2], [4]).

In this paper, we apply the linear structural equation model

* *Corresponding author*

to study the relationship between learning interests and their environmental influential factors in the learning and teaching of the course “Algorithms and Data Structures” in computer science.

The goal of our studies is to establish a linear structural equation on the relationships between learning interests and their environmental influential factors of the course “Algorithms and Data Structures”. By using the linear structural equation established we can analyze the influential factors of students’ learning interests and then provide a direction for improving teaching strategies in arousing and enhancing students’ learning interests of the course.

In the following 3 sections we describe our studies on the linear structural model to fit the influential factors of learning interests. In section 2 we discuss the linear structural equation model (LISREL) and its computing steps. In section 3 we study the linear structural model to fit the influential factors of learning interests.

2. The LISREL Model

Let x_1,x_2,...,x_q be q observable variables. These q observable variables have influences on the p observable variables y_1,y_2,...,y_p.

We can find n common factors (exogenous latent factors) ξ_1,ξ_2,...,ξ_n from the observable variables x_1,x_2,...,x_q and m common factors (endogenous latent factors) η_1,η_2,...,η_m from the observable variables y_1,y_2,...,y_p.

The problems to be analyzed will be the following.

Are there any relationships among the endogenous latent factors η_1,η_2,...,η_m?

Are there any relationships between the endogenous latent factors η_1,η_2,...,η_m and the exogenous latent factors ξ_1,ξ_2,...,ξ_n?

Let $Y = (y_1, y_2, \cdots, y_p)'$ be a vector of the dependent variables and $\eta = (\eta_1, \eta_2, \cdots, \eta_m)'$ be a vector of the endogenous latent variables.

Similarly, let $X = (x_1, x_2, \cdots, x_q)'$ be a vector of the independent variables and $\xi = (\xi_1, \xi_2, \cdots, \xi_n)'$ be a vector of exogenous latent variables.

The linear structural relation model (LISREL) consists of two parts. One part is a measurement model and the other part is a structural equation model.

The measurement model is used to describe the relationships between the endogenous latent vectors and the independent variable vectors, while the structural equation model is used to describe the relationships between the exogenous latent vectors and the dependent variable vectors.

2.1. The measurement model and the structural equation model

The measurement model can be formulated as

$$\begin{cases} Y = A\eta + \varepsilon \\ X = B\xi + \delta \end{cases} \quad (1)$$

The structural equation model can be formulated as

$$\eta = C\eta + \Gamma\xi + \theta \quad (2)$$

Where, A is a $p \times m$ coefficient matrix; B is a $q \times n$ coefficient matrix; C is a $m \times m$ coefficient matrix; Γ is a $m \times m$ coefficient matrix.

The vectors $\varepsilon = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_p)'$, $\delta = (\delta_1, \delta_2, \cdots, \delta_q)'$ and $\theta = (\theta_1, \theta_2, \cdots, \theta_m)'$ are all random error vectors.

2.2. Modeling steps

Let X be a vector of independent variables such as age, sex, score of foundation courses of the students and the teaching status of the teachers, Y a vector of dependent variables corresponding to 17 questions.

Modeling steps are as follows:

(1) Extract the exogenous latent vector ξ from the independent variable vector X and the endogenous latent vector η from the dependent variable vector Y respectively by using the exploratory factor analysis method.

(2) Build initial model based on professional knowledge; Estimate the parameters of the initial model using the maximum likelihood method; Compute the t value and the approximate standard error of the parameters; Assess the fitness of the model.

(3) Modify the initial model. Remove the variables not statistically significant step by step from the initial model. Build the final structural equation model according to the assessing variables computed.

(4) Compute the standard solution of the final structural equation model. Compute the standard solution of the parameters by transforming the covariance matrices of the latent vectors ξ and η to the corresponding correlation coefficient matrices.

(5) Draw the path graph of the standard solution of the final structural equation model. The path graph can intuitively reveal the structural relationships between ξ and η. The path graph consists of two parts. The observable variables X and Y are drawn in a rectangular box and the latent vectors ξ and η are drawn in an elliptical box. These two parts are connected by one-way arrows. The path coefficients are indicated in the connection lines.

The tasks of measurement modeling, structural equation modeling and estimating the variance and covariance matrices can be performed by programming the software package SAS ([1]).

The final model is assessed by the goodness of fit index (GFI) and the goodness of fit index adjusted for degree

(AGFI). The more close to 1 the two indices are, the better the fitness of the model ([5]).

3. The Linear Structural Model

We have performed an investigation of the undergraduate students in Fuzhou university about their learning interest and related factors. A total of 281 students were investigated. The content of the investigation consists of two parts. The part one of the investigation constituted by basic information of the students, basic information of students' scores of basic courses and the students' evaluation of teaching on the course "Algorithms and Data Structures".

The part two of the investigation is a learning interest scale constituted by 17 items noted as $y_1, y_2, \ldots, y_{17}$.

In the learning interest scale, there are 12 positive items and 5 negative items with "Yes" or "No" responses. For each positive items, a "Yes" response scores 1 and a "No" response scores 0. Contrarily, for each negative items, a "Yes" response scores 0 and a "No" response scores 1.

The Cronbach reliability coefficient of the learning interest scale applied to the 281 students is α=0.8465. This indicates that the scale can effectively measure the students' learning interest for the course "Algorithms and Data Structures".

3.1. Extraction of Latent Factors

We use the exploratory factor analysis method ([3]) to extract the exogenous latent vector ξ from the independent variable vector X and the endogenous latent vector η from the dependent variable vector Y respectively.

Four exogenous latent factors with corresponding eigenvalues grater than 1 were extracted from $x_1, \cdots, x_{19}$ by varimax rotation.

The variables with their loads not less than 0.5 were denoted as the main dominant variables.

The four exogenous latent factors extracted were: Knowledge basis of discipline (ξ_1), Knowledge demands (ξ_2), Course difficulty (ξ_3) and Teaching methods (ξ_4).

Similarly, three endogenous latent factors with corresponding eigenvalues grater than 1 were extracted from $y_1, y_2, \ldots, y_{17}$ by varimax rotation.

3.2. The Initial Model

From the result of the exploratory factor analysis, we can build the measurement model. According to the parameters in the maximum likelihood estimation model, the goodness of fit index of the resulting model is GFI=0.83. The goodness of fit index adjusted for degree is AGFI=0.80.

3.3. Modified Model

Since some parameters in the initial model have no significant difference with 0, it is necessary to optimize the initial model. We can get an improved model by removing the latent factors of $P > 0.05$.

The modified structural equation model can be obtained by using the generalized least squares method to estimate the standard parameters. In the modified model, the goodness of fit index is GFI=0.86 and the goodness of fit index adjusted for degree is AGFI=0.83. These two indices are both improved in comparison with the initial model.

From the modified model we can conclude that:

The impact factors of learning interest are:

Knowledge basis of discipline ξ_1, knowledge demands ξ_2, course difficulty ξ_3 and teaching methods ξ_4. The knowledge basis of discipline ξ_1 has a greatest impact, followed by the teaching methods ξ_4.

The impact factors of learning boredom are:

Knowledge basis of discipline ξ_1, course difficulty ξ_3 and teaching methods ξ_4. The course difficulty ξ_3 has the greatest impact, followed by the knowledge basis of discipline ξ_1. The course difficulty would be most likely to produce students' learning boredom. A good teaching method would reduce the students' learning boredom.

The impact factors of learning enthusiasm are:

Interest η_1, knowledge basis of discipline ξ_1, knowledge demands ξ_2 and teaching methods ξ_4. The interest η_1 has a greatest impact, followed by the knowledge basis of discipline ξ_1 and knowledge demands ξ_2. The teaching method ξ_4 has the smallest impact.

The students have good grades in the basic undergraduate courses such as mathematical analysis, linear algebra, discrete mathematics and programming languages have also a high learning interest and learning enthusiasm on the course "Algorithms and Data Structures".

In addition, the teachers' teaching skill is also an important factor on the learning interest and learning enthusiasm on the course "Algorithms and Data Structures".

An easy to understand teaching would promote the students' active learning and improve students' learning interest.

4. Concluding Remarks

We have studied the linear structural model fitting of the influential factors of learning interests.

Our results show that the main factors impacted on learning interest of the students are students' specialized basis and teachers' teaching skill. The weights of effect on learning interest for the two factors are 49.56% and 43.11%, respectively.

Teachers' teaching skill would promote the students' active learning and improve students' learning interest.

In addition, the knowledge basis of discipline has a direct impact on the students' study of the course "Algorithms and Data Structures". The students with good grades in the basic undergraduate courses have also a high learning interest and learning enthusiasm on the course "Algorithms and Data Structures".

References

[1] Cody Ron P. and Smith Jeffrey K.(2005) *Applied Statistics and the SAS Programming Language* (5th Edition), Prentice Hall.

[2] Duncan T. E., Duncan S. C. and Li F.(1998) A comparison of model- and multiple imputation-based approaches to longitudinal analysis with partial missingness. *Structural Equation Modeling*, 5, 1-21.

[3] Izenman Alan J.(2008) *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*, Springer-Verlag, New York.

[4] Marsh H. W.(1993) Stability of individual differences in multiwave panel studies: Comparison of simplex models and one-factor models. *Journal of Educational Measurement*, 30, 157-183.

[5] Mueller R O.(1996) *Basic Principles of Structural Equation Modeling: An Introduction to LISREL and EQS*. Springer-Verlag, New York.

[6] ACM/IEEE(2005) ACM/IEEE-Curriculum 2005 Task Force. Computing Curricula 2005, Computer Science. IEEE Computer Society Press and ACM Press, September 2005. (http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf)

Future of Digital Forensics: A Survey of Available Training

A. Evans, A. Williams, and J. Graham
Computer Science Department, Norfolk State University, Norfolk, VA USA

Abstract —*The field of forensics is multidisciplinary by nature, founded on the disciplines of criminology and information technology. However, most course work leading to a foundation in the field are offered within the Chemistry and Biology disciplines in addition to Computer Science. Regardless of the framework in which the courses are offered, the coursework itself must address social, legal, and ethical considerations for practitioners. In an effort to satisfy the different needs associated with forensics, there are different avenues one can pursue when seeking training in the field. Our research has identified 37 computer-related programs including some two-year associate degrees, four-year baccalaureate programs, graduate certificate programs, and non graduate certificate programs. First, we briefly define digital forensics. Second, we then discuss the different schools and what is unique about their programs. Third, we will discuss the certification programs that are available for those not interested in obtaining a degree.*

Keywords- forensics investigations; electronic discovery (eDiscovery); emulation learning (eLive); digital forensics; education

1. Introduction

As computer crimes have increased in numbers, so has the need for professionals within the field of digital forensics. The need to understand computer systems, how they operate, as well as how to attempt to keep them secure, remains of high importance. It is also important to understand computer forensics. After a computer system has been compromised and a crime has taken place, there is a need for a computer forensics investigation to follow. Computer forensics often gets mistaken for computer security and therefore anyone studying computer science can benefit from understanding not only the definition of computer forensics, but also its importance. Digital forensics differs from computer security which refers to securing computers against malicious attacks. It is also not the same as using computers to solve crimes. Criminal forensics techniques include matching fingerprints, ballistic testing, and DNA matching. By adding the ability to practice proper computer forensics techniques, it will help to ensure the overall integrity and survivability of the network's infrastructure. In addition, practicing good computer forensics techniques will help your organization, if you consider computer forensics as one of the basic elements in what is known as a "defense-in-depth" approach to network and computer security.

2. What is digital forensics?

Digital forensics is a branch of forensic science that was developed to meet an urgent need in the early 1990s. It consists of the recovery and investigation of material found in digital devices, often in direct relation to a computer crime. The term digital forensics was mainly used to refer to computer forensics but has expanded over the years to cover all devices capable of storing digital media and today is used to describe the entire field. National policies were developed in the early 2000s and now investigations can fall into one of four categories. Forensic analysis is probably the most common category. Within this category evidence is recovered to support or oppose a hypothesis before a criminal court. This could easily be confused with evidence gathering, where material is intended to identify other suspects/crimes. Then there is the electronic discovery (eDiscovery). This is a form of discovery related to civil litigation and intrusion investigation. eDiscovery is an investigation specifically designed to examine the extent of an unauthorized network intrusion. The technical side of investigations can be broken down into several sub-branches; computer forensics, network forensics, database forensics and mobile device forensics. An active investigation can utilize any number of the fields combined.

Computer forensic scientist or technicians are considered to be on the cutting edge within the criminal justice field. With the increase of cybercrime, cyber-terrorism, identity theft, and Internet child predators, computer forensic scientists are needed to track what were once thought to be traceless criminals. Computer forensic scientists are called on for many different agencies to be successful. Their expertise is needed for work with law enforcement officials, legal teams, independent companies and the government to conduct investigations, retrieve evidence and sometimes even testify in court. Computer forensic scientists remain in a high demand because of their thorough practical knowledge of computers, networks, hacking, data retrieval, information security, and computer surveillance. They are also perceived to be well trained in ethics as well as criminal justice topics like confidentiality, privacy laws and evidence handling. In most cases a computer forensics investigator will be required to work independently. There are some instances when forensic investigators will be under the direct supervision of a computer forensic scientist; however. forensic investigators should not only expect to work independently, they should also be prepared to work under stressful situations. Digital forensics is not limited to identifying direct evidence of a crime, it can also be used to link evidence to specific suspects, confirm alibis or statements, determine criminal intent, or identify sources in copyright cases. With digital forensics, the investigations are broader in scope than other areas of forensic analysis. The digital forensic process is made up of the seizure, forensic imaging and analysis of digital media. Once all is said and done, there is a report produced documenting the digital evidence for the courts or an employer.

3. Degree Training Programs Offered

The research performed for this paper uncovered 38 computer forensics-related programs including two-year associate degrees [7, 11, 14, 15, 18, 21, 22], four-year baccalaureate programs [3, 5, 7, 8, 26, 38], master's degree programs of study [11, 25, 26, 32], graduate certificate programs [10, 23, 24, 32, 33] and non-graduate-level certificate programs [1, 2, 4, 6, 9, 13, 16, 17, 19, 20, 25, 27, 28].

3.1 Associate Degree Programs

In an effort to satisfy the different needs associated with forensics, there are different avenues one can pursue when seeking training in the field. At the City College of Chicago [37], they offer several programs leading to an Associate in Applied Science degree (A.A.S.) for information Technology. They also offer a Basic Certificate in Computer Security and Forensic Investigation (CSFI). Their CSFI program was particularly interesting in that it covered two areas of interest, Information Security and Computer Forensic Law Enforcement. The Information Security area of emphasis focuses on the design, implementation and management of the information security in the corporate environment. This course prepares their students for the Certified Information Systems Security Professional (CISSP) Exam. The CISSP certification is nationally recognized. The area of emphasis that focuses on Forensic/Law Enforcement focuses mainly on the computer forensic investigation and provides law enforcement personnel, criminal justice majors with procedures and methods for investigation of computer crimes and handling electronic evidence.

3.2 Baccalaureate Programs

Of the four-year colleges researched, Defiance College was the one that seemed to offer the most in-depth training. Their program offers unique opportunities for students to study and gain hands-on experiences working in labs, and mock crime-scenes. The tools and techniques of digital forensics are also applied in situations where data is constantly in motion, such as while recognizing and responding to intrusions into a company's computer network, or when recovering data from small-scale digital devices such as cell phones and PDAs. Of all the evidence collecting procedures digital evidence collection is probably the most tedious process requiring the most discipline. It is important for the evidence to be precise for it to be admissible in court. Here the students learn how to preserve the integrity of digital evidence; extract live, static, and deleted data from various media; and thoroughly document and present their findings. In addition, students in this program develop a well rounded background which consist of general education, criminal justice, and computer technology fundamentals. At the completion of this program the students will graduate with two professional certificates in hand, a degree in Digital Forensic Science, and the practical experience gained from utilizing the internship program. [38]

First, the students learn about the computers by preparing for the CompTIA A+ certification as an IT Technician. This certification is widely respected when seeking entry level information technology employment. Next the students begin learning about operating systems, security principles, and networking. Once the computer background is established the students begin learning the science of forensics and how it is applied the digital aspect of computers that are turned off, computers that are still running, computers communicating with one another, and small-scale digital devices such as cell phones, PDAs, smart phones, and other hybrid systems. In addition, the students learn the value and importance of having high ethics and personal integrity. After learning all of these skills sets the student have the opportunity to fine tune their skills through an internship with one of many different agencies performing forensics work.

The different internships offered to students will be afforded to them in their senior year. These are opportunities for the students to perform hands-on training and for them to participate in actual cases that are in the process of being prosecuted. The courses offered at Defiance College in Digital Forensics are as follows: CompTIA A+ exam prep, Introduction to Computer and Digital Forensics, Computer Security Fundamentals, Operating Systems, Computer Forensics and Security Ethics, Law Enforcement Field Experience, Seizure and Forensic Examination of Computer Systems, Advanced Topics in Computer Data Analysis and Recovery, Fundamentals of Computer Networks, Network Forensics, Intrusion Detection, National Certification, and Computer Forensic Field Experience and Seminar.

3.3 Graduate Programs

When dealing with law enforcement, computer forensics scientist and investigators are considered to be highly educated and are therefore expected to have a bachelor's degree. Because of the limited programs available, obtaining a Master's degree in a field like computer science or criminal justice can be difficult. Once obtained, the master's degree can prove to be extremely beneficial in acquiring the top level jobs. Students in these programs will learn about cutting edge technologies, systems and concepts needed to succeed in computer forensics. Those with Master's degrees can also expect to receive a higher salary than those who only hold undergraduate degrees, and they may be promoted earlier and more frequently also. Each of the Graduate Degree programs differ slightly in the courses required to complete the degree. All of the programs researched required between 33–40 semester credits to complete the degree [11.25,26,32] .

3.4 Certificate Programs

Computers and various digital devices are essential in the daily operations of organization. This high dependency of technology brings with it serious security challenges. Criminals make their living hacking into these businesses. In order to mitigate these risks there must be a way for IT professionals to obtain a comprehensive overview of digital forensics. Of the certificate programs we researched, Boston University's Graduate Certificate program was most appealing. This specialized graduate certificate program in digital forensics provides the students with comprehensive "digital crime scene investigation" knowledge. The program introduces students to forensic analysis policy and procedures, forensic analysis tools, data recovery, and investigation, among other topics. The program is unique in that it offers the course work on campus and via an online format called Emulation Live format (eLive). This format allows the students to get a blend of both on campus classroom sessions as well as the online courses.

eLive courses include:

- Several traditional, face-to-face classroom sessions throughout the semester
- Online content that allows students to complete coursework and collaborate with classmates and instructors using Internet service
- Multimedia online technology utilizing virtual lectures, video conferencing, real-time collaborative sessions, correspondence, projects, and assignments
- Course materials and discussion threads that are accessible online 24 hours a day, 7 days a week

4. Uses for Degree Training

The majority of Information Technology (IT) professionals in business and law enforcement agencies already have college degrees. They do not need to concern themselves with the programs leading to a two or four-year degree. In these cases a certificate program would be more advisable than a second college degree to perform computer forensics works. Now that the need for digital forensics experts is on the rise, students are encouraged to pursue a degree program of some sort. More businesses are looking for people with specialized skill sets. Digital forensics is a field that requires attention to detail in all aspects of the job. Digital forensics jobs are often found within law-enforcement, military, government intelligence agencies and private security or consulting companies. To get a more in-depth idea of the types and number of jobs available you can search a popular job bank called Dice and it will return around 160 different jobs at the time of publication. If you do the same thing with Monster.com you will return around 210 different jobs. The numbers are not staggering, but they are on the rise. The different job titles varied from Computer Forensics Analyst to Vulnerability Security Research Engineer. Most of these jobs require a degree of at least two years of experience and others require a security clearance. Some of the jobs would accept equivalent knowledge and job experience in place of the required education, or just the opposite, an advanced degree in place of some of the required experience.

4.1 Salary Ranges

It has been predicted by the Bureau of Labor Statistics that computer forensic investigators will remain in high demand for the next several years to come. The data provided covers both self employed forensic investigators as well as those employed by a firm. Generally speaking there will be more stability offered by a firm. Self employed investigators would enjoy more flexibility. Their salary could also be either substantially lower or higher than their counter parts working within a firm. When analyzing the data we found that digital forensics is usually combined or closely related to the criminal justice field. Employment of private detectives and investigators is expected to grow 22 percent over the 2011–21 decade, substantially faster than the average for all occupations [39]. As mentioned previously in this paper, the increased demand for private detectives and investigators is a direct result from the heightened security concerns, increased litigation, and the need to protect confidential information and property of all kinds. Criminal activity on the internet includes spamming, identity theft, e-mail harassment, and more recently illegal downloading of copyrighted materials. The average salary reported in 2006 by the Bureau of Labor Statistics was $33,750 for private investigators[39]. Generally those in the field of computer forensics earn higher salaries. “Median annual wages of salaried private detectives and investigators were $41,760 in May 2008. The middle 50 percent earned between $30,870 and $59,060. The lowest 10 percent earned less than $23,500, and the highest 10 percent earned more than $76,640. Wages of private detectives and investigators vary greatly by employer, specialty, and geographic area.”[39] Depending on the casework, these professionals may also encounter irregular schedules and long overtime hours. Related Works

There is currently ongoing research targeted at enhancing the way forensic courses are being taught. The Cyber Defense Trainer (CYDEST) [34] was developed by a group of researchers to incorporate the use of a virtualized training platform for computer forensics. Through the use of virtual machines, it provides tactical level exercises for network administrators, first responders, and digital forensics investigators. CYDEST is more than just a tool to create different scenarios, it also makes it possible for student’s every action to be properly monitored and recorded for later viewing by their professors. CYDEST ultimately provides a level of realism and automation for the student while reducing the workload of an instructor [34]. In addition, there is other research that argues the best form of realism is the real thing. In their paper, “Designing Computer Forensics courses using Case Studies to Enhance Computer Security Curricula” [35], they argue that computer forensics courses should be designed and taught utilizing actual court cases that represent actual cyber crimes. They believe that by using actual cases the students will learn more and better understand the existing Cyber Laws. Students can perform their own forensic investigations and compare them with the actual case and learn from their mistakes or reinforce the positive. There is also research being done in the online arena. Wang, of Southern Polytechnic State University, has written a paper, “Web-Based Interactive Courseware for Information Security”, that talks about teaching such topics as security and forensics, utilizing Web-based multimedia and interactive courseware. The courseware is based on the use of a tool called Multimedia and Interactive Courseware Synthesizer, (MICS) [36]. MICS provides an interactive platform for students to learn. Security courses in general can be hard to master without student instructor interaction. MICS, on the other hand, provides an interactive learning environment that can also incorporate games to be used for instruction. The primary focus of MICS is to be able to develop a collection of hands-on labs that cover security, privacy, reliability and business integrity.

5. Conclusion

In conclusion, it is evident that the need for computer forensics specialists is increasing. As this need continues to rise, it is imperative that educational opportunities increase as well. In addition, we must strive to stay on the cutting edge of the industry’s demands. Digital forensics is a new and a constantly changing field which requires constant improvement and updating. There has to be more of an effort to consult practitioners in the field in order to continue developing relevant course material. There should also be a greater effort towards encouraging more organizations to offer internships in the field of digital forensics. In addition, we found that computer forensics is a growing multi-disciplinary field with increasing industry demand. Even with the lack of publicly recognized program standards to follow at this time, there are a number of good examples of two and four year programs [21, 22, 38]. At the master’s level, [26, 32] serve as good examples. In comparison to other Scientific Fields of study, few programs exist at this time for computer forensics. Most of the programs are located in eastern portion of the United States with about 25% of the programs on the west coast. It is evident that there is plenty room for future growth with respect to the training programs offered in the Digital Forensics field

6. References

[1] http://www.blueridge.edu/Continuing/Education/Programs/CyberCrimeInvest.htm.

[2] http://cte.bridgew.edu/certifications/cdfe.cfm.

[3] http://programs.bcit.ca/845CBTECH.

[4] http://web.bryant.edu.

[5] http://www.bc3.edu/academics/technology/compforensics.htm.

[6] http://canyoncollege.edu/lawenforce.edu.htm.

[7] http://digitalforensics.champlain.edu.

[8] http://a-s.clayton.edu/cj/curriculum.

[9] http://www.curry.edu/Academics/Continuing+Education/Certificate+Programs/Computer+Crime_Investigations+and+Computer+Forensics.htm.

[10] http://www.gwu.edu/~mastergw/programs/com_fraud.

[11] http://www.icmschool.com/business-medical-careers-criminal-justice-cybercrime.html.

[12] http://www.jjay.cuny.edu/programsGraduate/progGraduateForensicComputing.asp.

[13] http://www.kennesaw.edu/coned/sci/index.htm

[14] http://www.lwtc.ctc.edu/future/programs/list/cfor.htm.

[15] http://www.lccc.edu/academics/credit/computer-forensics-aas.asp.

[16] http://www.bus.oregonstate.edu/services/nti.htm.

[17] http://www.polk.edu/INSTRUCT/wfd/ips/programscoursedescriptions.htm.

[18] http://www.southwest.cc.nc.us/acadprog/cct.htm

[19] http://tech.spokanefalls.edu/InfoSys/default.asp?menu=2&page=CertForensics.

[20] http://www.spcollege.edu/webcentral/acad/compcrime.htm.

[21] http://starkstate.edu/ACADEMICS/it_tech/compnetadmin_secur.htm.

[22] http://www.sunytccc.edu.academic/forensics/main.asp

[23] http://www.graduate.ucf.edu/currentGradCatalog/content/degrees/ACAD_PROG_71.cfm.

[24] http://www.newhaven.edu/psps/gradforensicscience.html.

[25] http://www.extension.washington.edu/ext/certificates/cpf/cpf_gen.asp.

[26] http://www.ecii.edu/edu_eci.html.

[27] http://www.lcsee.cemr.wvu.edu/forensics/.

[28] http//wright.ccc.edu/department/forensics/index.asp

[29] J. Heiser, and W. Kruse, “What Exactly is Computer Forensics?”

[30] http://wwww.netsecurity.com/forensics/digital_computer_forensics_services.html.

[31] http://www.business.latech.edu/graduate/iac_courses/CIS522.pdf.

[32] http://www.2009-2009.graduatecatalog.ucf.edu/Programs/Programs.aspx

[33] http://www.bu.edu/met/programs/graduate/digital-forensics-certificate/

[34] Bruekner, S., Guaspari, D., Adelstein, F., & Weeks, J. (2008). *Automated computer forensics training in a virtualized environment*. Retrieved March 7, 2011, from Scince Direct:

http://www.dfrws.org/2008/proceedings/p105-brueckner.pdf

[35] Herath, A., Herath, S., Goonatilake, R., Herath, S., & Herath, j. (2007). “Designing computer forensics courses using case studies to enhance computer security curricula.” *Journal of Computing Sciences in Colleges*, 264-271.

[36] Wang, A. J. (2005). “Web-Based Interactive Courseware for Information Security.” *SIGITE '05 Proceedings of the 6th conference on Information technology education* (pp. 199-204). New York: ACM.

[37] http://www.ccc.edu/

[38] http://www.defiance.edu/page/major_Digital_Forensic_Science.html

[39] http://www.bls.gov/oco/ocos157.htm#outlook

Test Time vs. Test Performance

N. Tadayon[1], **C. Nyman**[1], **and N. Barker**[1]
[1]Department of Computer Science and Information Systems
Southern Utah University, Cedar City, Utah, USA

Abstract *- This paper analyzes the relationship between test time and test performance (score) based on gender, delivery type, and classification of students within two tests. The course being analyzed is a computer literacy course that is part of the required general education curriculum. This course is divided into two sections, Applications and Technology. The Applications section of the course has specific outcomes that are defined by the standards set by state computer literacy requirements. The topics used in the Technology section of the course include more up-to-date basic knowledge of computer software which is based on the results of the students' surveys and faculty input with the primary objective being to expose students to some of the necessary dynamic technological competencies. This paper analyzes the time vs. score for the two tests within the Technology section of the course.The tests are comprised of several multiple choice questions taken online. The result of the analysis shows that although the performance of students increased based on the time they spent on the test, there are differences among gender, students' standing, and delivery type.*

Keywords: Computer literacy, Technology education, Comparison, Application software, Test time, Test performance.

1. Introduction

This paper concentrates on analyzing the performance of students vs. the time taken to complete testing. In this paper, test performance is evaluated by test scores. Since the "Intro. to Computer Apps & Internet" (as part of the required Computer Information Literacy (CIL) for all students) course is used for this study, it seems relevant to initally examine the course itself.

The term "computer competency" is often used by educational institutions to define the computer knowledge and skill requirements or as a proficiency standard for their graduates. The computer competency requirement is primarily identified at the State level, with various interpretations among institutes within the State. For example Florida State University (http://facsenate.fsu.edu/Liberal-Studies-Competency-Forms/Computer-Competency) has a computer competency requirement based on a course for their students to demonstrate:

1. Competent use of a *discipline-useful software package*
2. The ability to perform simple transactions using the web/Internet.

Southern Utah University (SUU) offers a required course to meet the "Information Literacy, Computer Literacy, and Student Success" requirement within their General Education. Suggested competencies within the "Information Management" category include demonstrating proficiency in the use of computer operating systems, databases, word processing, and spreadsheets and their applications in major fields of study. However, CIL required knowledge in Utah State University includes six exams in Information Law and Ethics, Information Resources, Document Processing, Computer Systems, Spreadsheets, and Electronic Presentations. The Computer Competency requirement within the secondary educational systems within Utah is defined within the IC^3 exam (Internet and Computing Core Certification). IC^3 is designed "to assess skills in three main areas: computing fundamentals (hardware, software and operating systems), key applications (common functions, word processing, spreadsheets, and presentation software), and living online (networks and the internet, email, using the internet, and the impact of computing and the internet on society). A set of technology standards for K-12 grade levels was developed by ISTE (The International Society for Technology in Education) that included basic operations and concepts, technology productivity tools; and technology communications tools". (ISTE, 2011).

There is no denying the increasing need for computer competency among higher education graduates in order to boost their effectiveness and productivity in tomorrow's workforce. The term "computer literacy" extends to a wider view than computer usage. The relatively new term "*Computer Fluency"* contains computer literacy but adds a broader coverage of technological topics to prepare students to keep pace with ever-advancing technology. To meet the technological demands of the 21st century, SUU has included a section related to technology for graduates to the CIL course. The tests within the technology sections will be analyzed. At SUU, we have developed a formative evaluation to facilitate the improvement of a computer literacy course in order for the course offered to correspond with the needs of the students, aligned with technological advances and employer requirements through surveys. This course is driven by hands-on projects and in-class activities which have been proven to be the most helpful strategies [4].

2. Technology coverage at SUU

The technology section of the “Introduction to Computer Apps & Internet” course, as part of the computer literacy requirement at SUU, is organized in two sections, as follows:

Technology 1 includes the following topics:

- Firefox and extensions
- E-Commerce
- Audio Editing
- 3D Computer Graphics
- Cloud Computing
- Linux and Servers
- Video Editing

Technology 2 contains the following topics:

- Networking
- Web Development
- Photo Editing
- Cyber Crime
- Open Office
- Blogs

To compare and analyze the relation between time and performance within two tests for the technology section, the students are classified based on gender (male and female), classification (freshmen, sophomore, junior, and senior), and delivery method (in-class and on-line). There is a study by Zoran Sevarac on modeling students using neuro-fuzzy systems which classifies students based on test results and the time spent testing as Bad, Good, Very Good, and Excellent using the whole population of students. (Sevarac, 2006)

The relationship between test completion time and test performance is a topic that has been explored previously in literature, with mixed outcomes. (Foos, 1989; Paul, 1980; Wierzbicki, 1994; Bridges, 1985; Feinberg, 2004; Terranova, 1972; Landrum, 2009). Three possible relationships between completion time and test performance were proposed: (a) a linear relationship where students who complete exams more quickly score better; (b) a curvilinear relationship where students who have close to average completion times score better; and (c) a relationship where average completers exhibit less test score variability compared to fast and slow completers. The results in our test have been more or less consistent and indicate better scores for more time taken in the test. However, the slope of the linear trend line for each category is different, which contradicts some of the previous findings.

A total of 730 students took the Computer Literacy course during the Fall 2010 semester in 13 in-class and 10 on-line sections. The average enrolment was about 32 students per section, but about 20% of the students did not take or went over the time limit for the tech1 or tech2 tests. This means that there were 149 students who were not included in our data analysis. The demographic of data is shown in table1.

	Freshman	Sophomore	Junior	Senior	Total
Male In-Class	70	50	12	6	138
Male On-line	38	36	12	13	99
Female In-Class	128	64	14	6	212
Female On-Line	46	65	12	9	132
Total	282	215	50	34	581

Table 1 Demography of students

Figures 1 and 2 show the percentage of student who took this course in-class and on-line separated by student standing. It was interesting to note that more freshman students took this course in-class and senior students took this course on-line.

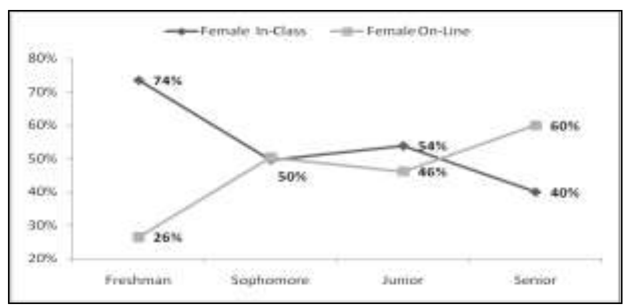


Figure 1 Percentages of Female In-class vs. On-line

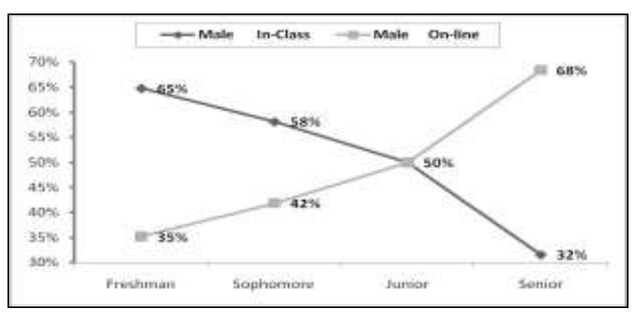


Figure 2 Percentage of Male In-class vs. On-line

It is interesting to note that freshman students tend to take this course in-class and seniors on-line.

3. Results

The analysis was done with 581 students for two tests within the technology sections of the course. The test questions were all identical multiple choice questions shown in random order, and the test was taken online. The time limit for the tests was 50 minutes and the highest possible score for each test was 40 points. Questions were delivered one at a time and revisiting a question was not allowed. The trends in the results of the test were compared against the time spent testing for all students and were based on gender, classification, and delivery mode. The linear trend line as well as a second degree polynomial trend curve are displayed and compared.

The results of the trend in time versus performance in the two technology section tests of the course are within the following subsections.

3.1 Test1

Figure 3 contains the overall results for test1.

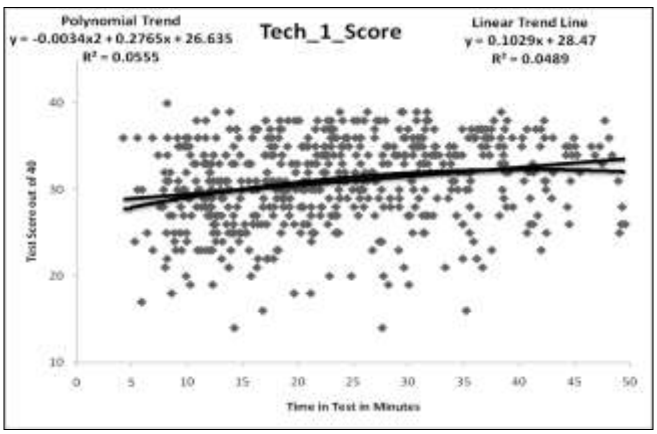


Figure 3 Time vs. score in Test1 (Overall)

The average score for test1 was 30.89/40 = 77.22% with a standard deviation of 4.91, and the average time on test1 was 23.53 minutes with a standard deviation of 10.55. The linear trend line for test1 was y = 0.1029x + 28.47 with R^2 = 0.0489. This demonstrates a slight increase in score for students who spent more time. The data are significantly scattered, and R^2 = 0.0489 shows this fact. However, the polynomial trend for test1 was y = -0.0034x2 + 0.2765x + 26.635 with R^2 = 0.0555, which shows that overall the students who spend approximately 40.66 minutes had the optimal score, although the average time was 30.89 minutes.

3.1.1 Results in test1 based on gender

Figures 4 and 5 show the results of test1 divided by gender.

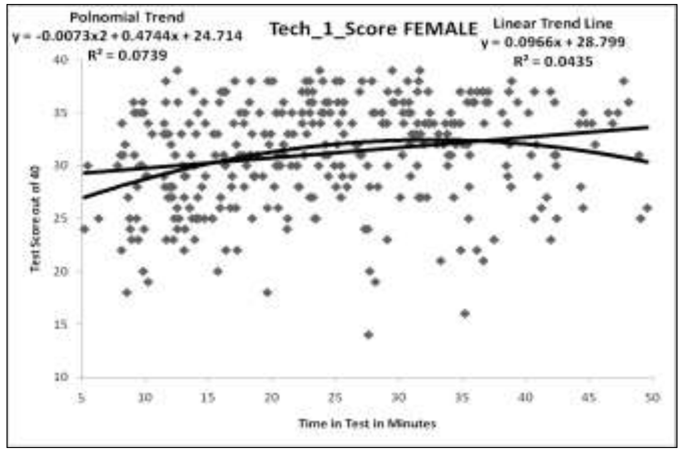


Figure 4 Time vs. Score in Test1 (Female)

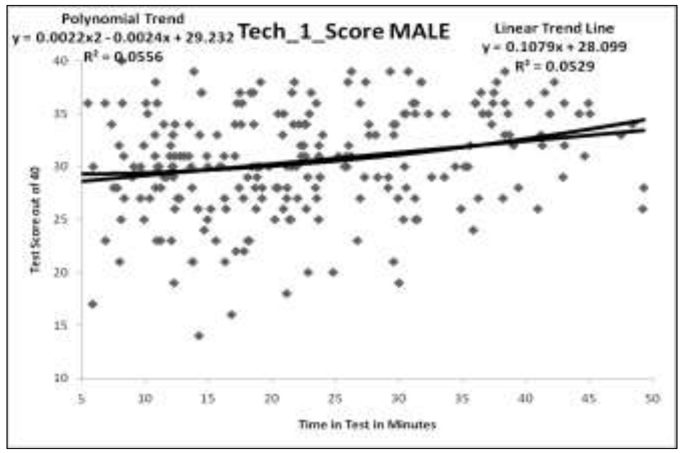


Figure 5 Time vs. Score in Test1 (Male)

There were 344 female students, and their average score for test1 was 31.14/40 = 77.84% which is slightly higher than the overall average with a standard deviation of close to 4.85.

The number of male students was fewer (237 students) and their average score 30.53/40 = 76.33% was slightly less with a standard deviation of 4.99.

The average time in test1 for female students was 24.20 minutes with a standard deviation of 10.45. The average time in test1 for male students was 22.54 with a standard deviation of 10.64. This difference shows that overall, female students spent more time on test1 than males and had a better overall average in their test1 score.

The linear trend line for female students in test1 was y = 0.0966x + 28.799 with R^2 = 0.0435, and the linear trend line in test1 for male students was y = 0.1079x + 28.099 with R^2 = 0.0529. Although both female and male students increased their score based on time, it is clear that the linear trend for the male students has a slightly higher slope. However, the polynomial trend for female students taking test1 was y = -0.0073x2 + 0.4744x + 24.714 with R^2 = 0.0739, and the polynomial trend for male students was

$y = 0.0022x2 - 0.0024x + 29.232$ with $R^2 = 0.0556$. This indicates a difference in trends and data. The female students who spent 32.5 minutes had the overall optimal score (relative to the time spent on the test) with a relatively high curvature in the graph whereas the male students did not demonstrate the same trend. The male students had an opposite curvature and consistently (within time in the test) had a better score with more time spent.

3.1.2 Results in test1 based on students' standing

Figure 6 shows the results of test1 divided by classification (Freshman, Sophomore, Junior, and Senior).

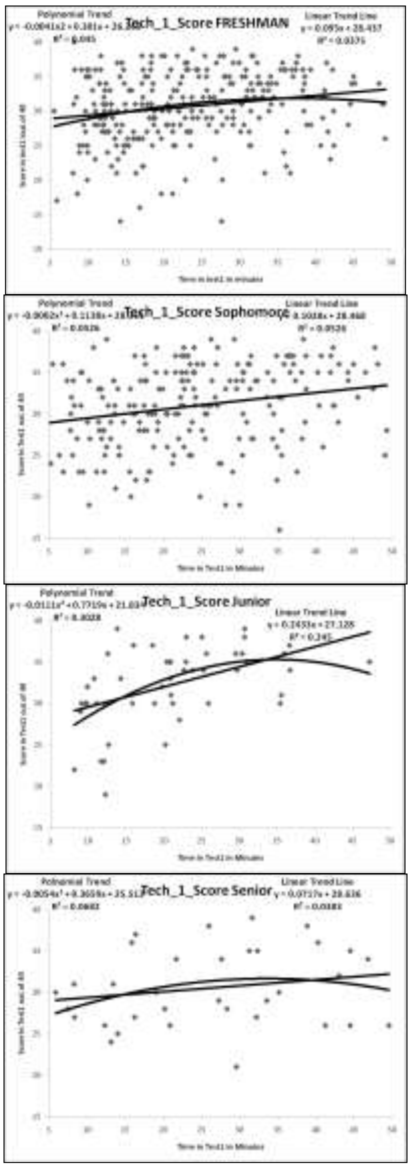


Figure 6 Time vs. Score in Test1 (Freshman, Sophomore, Junior, Senior)

This course is an introductory course and most of the students were freshmen and sophomores. The linear trend lines and polynomial trend for the time vs. score in test1 are shown in Table 2.

Classification	Linear Trend Line	Polynomial Trend
Freshman:	$y = 0.095x + 28.43$ $R^2 = 0.0375$	$y = -0.0041x^2 + 0.301x + 26.265$ $R^2 = 0.045$
Sophomore:	$y = 0.1028x + 28.46$ $R^2 = 0.0526$	$y = -0.0002x^2 + 0.1138x + 28.351$ $R^2 = 0.0526$
Junior:	$y = 0.2433x + 27.12$ $R^2 = 0.245$	$y = -0.0111x^2 + 0.7719x + 21.834$ $R^2 = 0.3028$
Senior:	$y = 0.0717x + 28.63$ $R^2 = 0.0383$	$y = -0.0054x^2 + 0.3659x + 25.512$ $R^2 = 0.0682$

Table 2 Linear and Polynomial Trend Based on Students' Classification on Test1

The average time and score in test1 for each category is shown in Table 3.

Classification	Average Time	Average Score
Freshman	23.10	30.63
Sophomore	24.10	30.94
Junior	21.50	32.36
Senior	26.39	30.53

Table 3 Average Time and Score Based on Students' Classification on Test1

It is noteworthy that the average score in test1 is highest among junior students, who spent the shortest time on the test; and the lowest average score was among senior students, with the highest average time spent. There were only 50 junior and 34 senior students out of 581 students in the data taking test1.

Based on the data, there were 497 freshman or sophomore students (over 85.5% of students) taking test1; and the polynomial trend indicates that the best time vs. score for freshman students was 36.7 minutes, whereas the sophomore students showed a different trend, with a consistently small decrease in slope. The best time vs. score for juniors and seniors was 34.77 and 33.88 minutes, respectively.

It is also interesting to compare the portion of female students in each category. Based on data, 61.7% of freshman students, 60% of sophomore students, 52% of junior students, and 44% of senior students were female.

3.1.3 Results in test1 based on delivery method

Figure 7 shows the compared result of students on-line and in-class.

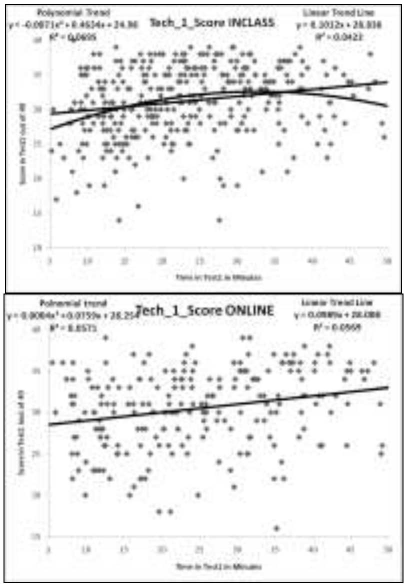


Figure 7 Score vs. Time in Test1 (In-class and On-line)

There were 350 students that took this course in-class and 231 on-line. The average time and score for on-line and in-class students are as follows:

Delivery Method	Average Time	Average Score
In-Class	22.94	31.16
On-Line	24.76	30.46

Table 4 Average Time and Score Based on Delivery Method on Test1

As shown in Table 4, the in-class students have higher average scores in test1 with lower average times compared to on-line students. The in-class students have a slightly higher slope in the linear trend line of $y = 0.1012x + 28.836$ with $R^2 = 0.0422$ compared to the on-line students with the linear trend line of $y = 0.0989x + 28.008$ with $R^2 = 0.0569$. The percentage of female students was approximately 61% within in-class and 57% within on-line classes. The polynomial trend for in-class students was $y = -0.0071x^2 + 0.463x + 24.98$ with $R^2 = 0.0695$. This demonstrates that the best ratio of scores was within 32.63 minutes. However, the polynomial trend for on-line students was $y = 0.0004x^2 + 0.0759x + 28.254$ with $R^2 = 0.0571$ which indicates an overall slight increase in slope of scores compared to time.

Approximately 70% of freshman students, 53% of sophomores, 52% of juniors, and 35% of seniors took the course in-class.

3.2 Test2

Figure 8 contains the overall results for test2.

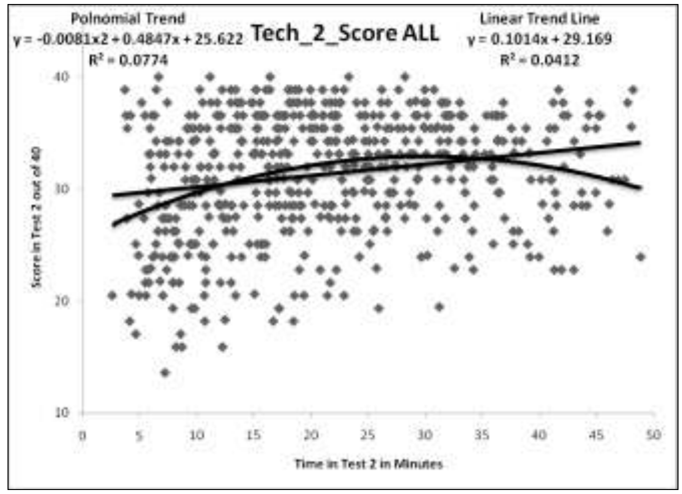


Figure 8 Time vs. Score in Test2 (Overall)

The average score for test2 was 31.32/40 = 78.3% with a standard deviation of 5.42, and the average time on test2 was 21.17 minutes with a standard deviation of 10.85. The average score in test2 was higher than the one in test1, and the average time on the test is slightly lower. The linear trend line for test2 $y = 0.1014x + 29.169$ with $R^2 = 0.0412$ has a similar slope compared to test1. The polynomial trend for test2 was $y = -0.0081x2 + 0.4847x + 25.622$ with $R^2 = 0.0774$ which shows that the peek of increase in score with respect to time was at 29.92 minutes, demonstrating a change compared to test1 which was 40.66 minutes.

3.2.1 Results in test2 based on gender

Figure 9 shows the overall result of test2 based on gender.

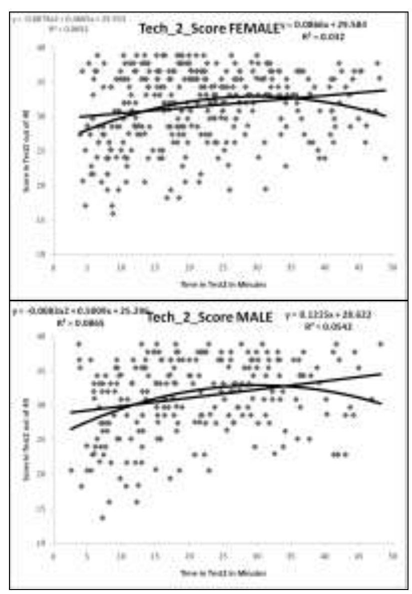


Figure 9 Time vs. Score in Test2 (Female and Male)

The average in time and score is shown in Table 5.

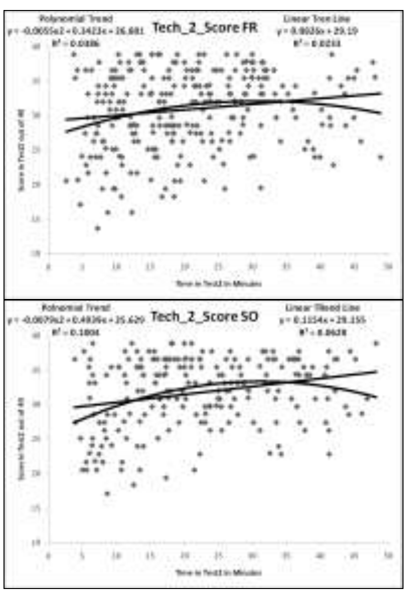
Gender	Average Time	Average Score
Female	21.61	31.45
Male	20.54	31.12

Table 5 Average Time and Score Based on Gender on Test2

The average time for both genders decreased, and the average score for both genders increased compared to test1 results. However, the linear trend line for test2 within female students was y = 0.0866x + 29.584 which indicated a slight decrease in slope whereas the linear trend line for test2 within male students was y = 0.1215x + 28.622 which showed a slight increase in slope compared to test1. The main difference may be clearer from comparing polynomial trends between the two tests. The polynomial trend for test2 within female students was y = -0.0078x2 + 0.4665x + 25.933 which indicated the highest change in slope of the linear trend line at 29.9 minutes. This may be considered a similar trend to female students taking test1. However, the male students taking test2 had a polynomial trend of y = -0.0083x2 + 0.5009x + 25.296 which differs from test1 and showed an optimal point in the graph at 30.17 minutes.

3.2.2 Results in test2 based on students' standing

Figure 10 shows the results of test2 divided by classification (Freshman, Sophomore, Junior, and Senior)



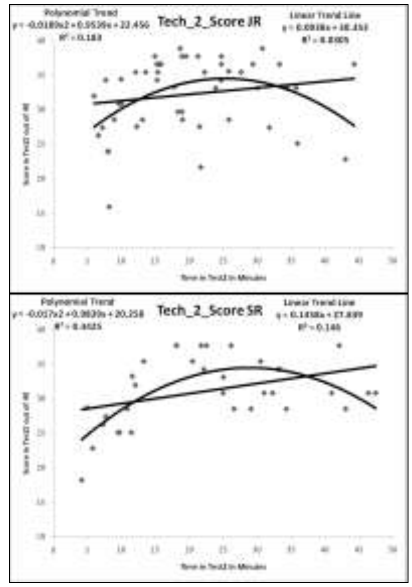


Figure 10 Time vs. Score in Test2 (Freshman, Sophomore, Junior, Senior)

The linear trend lines and polynomial trend for the time vs. score in test2 are shown in Table 6.

Classification	Linear Trend Line	Polynomial Trend
Freshman:	y = 0.0826x + 29.19 R² = 0.0233	y = -0.0055x2 + 0.3423x + 26.801 R² = 0.0386
Sophomore:	y = 0.1154x + 29.15 R² = 0.0628	y = -0.0079x2 + 0.4939x + 25.629 R² = 0.1004
Junior:	y = 0.0938x + 30.35 R² = 0.0305	y = -0.0189x2 + 0.9539x + 22.456 R² = 0.183
Senior:	y = 0.1458x + 27.84 R² = 0.146	y = -0.017x2 + 0.9839x + 20.258 R² = 0.4425

Table 6 Linear and Polynomial Trend Based on Students' Classification on Test2

The average time and score in test2 for each category is shown in Table 7.

Classification	Average Time	Average Score
Freshman	20.62	30.89
Sophomore	21.95	31.69
Junior	19.85	32.22
Senior	22.85	31.17

Table 7 Average Time and Score Based on Students' Classification on Test2

This data supports the previous conclusion on test1, that the highest test score and lowest time belonged to junior students. However, since the number of juniors was limited to 50, which is less than 10% of all students, the findings cannot be conclusive.

3.2.3 Results in test2 based on delivery method

Figure 11 shows the results of in-class vs. on-line students.

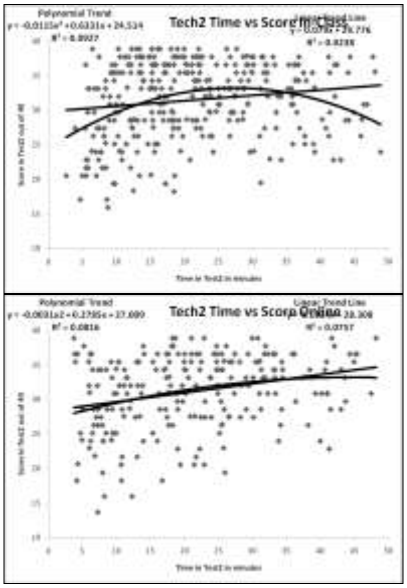


Figure 11 Time vs. Score in Test2 (In-class and On-line)

The average time and score for on-line and in-class students in test2 are shown in Table 8.

Delivery Method	Average Time	Average Score
In-Class	21.12	31.45
On-Line	21.26	31.12

Table 8 Average Time and Score Based on Delivery Method on Test2

A quick comparison with test1 results showed that the average time for on-line students decreased by less than 14% whereas the average score was slightly higher. The trend line was similar to test1 but the polynomial trend for in-class students was much sharper, indicating that the change in slope in the linear trend line of time vs. score was much sharper for test2.

The linear trend line for in-class students in test2 was $y = 0.079x + 29.776$ and had less slope compared to in-class on test1, which was $y = 0.1012x + 28.836$. The polynomial trend showed that the optimal point in test2 was 27.5 minutes compared to test1 which was 32.63 minutes. The on-line class students in test2 had a linear trend line of $y = 0.1324x + 28.3$ which, compared to test1 ($y = 0.0989x + 28.008$), showed a higher slope in line. The polynomial trend in test2 had a maximum point at 44.9 minutes compared to test1 which was an overall slight increase in slope (descending). This may be due to the types of questions within the two tests.

4. Conclusion

The similarity between all students within test1 and test2 was within their linear trend lines. All data supported a positive trend in students getting a better score as the time spent testing increased even though the optimal time (the minutes with maximum point for the 2nd degree polynomial trend) was different. It also implied that on average, the junior students spent the least amount of time and had the highest scores on both tests. Most freshman students (both female and male) took the course in-class, and most senior students (both female and male) took the course online. However, in-class students spent less time taking the test and earned higher scores.

5. References

[1] Easton, Annette C., George Easton and Theo Addo. "But I Am Computer Literate: I Passed The Test.", Journal of College Teaching & Learning (Feb 2006, Vol 3, No 2): 39-44.

[2] R. E. LANDRUM, H. C. "The Relationship Between Time to Complete a Test and Test Performance.", *Psychology Learning & Teaching*, 53-56. V 8.2 (2009)

[3] Courte, J. E. "Comparing Student Acceptance and Performance of Online Activities ro Classroom Activities." SIGITE'07 (pp. 185-189). Destin, Florida, USA: ACM. (2007).

[4] Robbins, R., & Zhou, Z. "A COMPARISON OF TWO COMPUTER LITERACY TESTING APPROACHES.", Issues in Information Systems , pp. 185-191. Vol. VIII, No. 1 (2007)

[5] Daughenbaugh, R., Daughenbaugh, L., Surry, D., & Islam, M. "Personality Type and Online Versus In-Class Course Satisfaction.", Educause, pp. 71-72 (2002)

[6] Gondhalekar, V., Edwards, S., & Barnett, R. "Online vs. In-Class: Comparison Based on Grades.", Financial Management Association (FMA). (2007)

[7] Kleinman, J. a. "Comparison of In-Class and Distance-Learning Students' Performance and Attitudes in an Introductory Computer Science Course.", The Journal of Computing in Small Colleges , 206-219. (2002)

[8] Liao, L., & Pope, J. W. "COMPUTER LITERACY FOR EVERYONE.", Consortium for Computing Sciences in Colleges, (pp. 231-238). (2008)

[9] Parkhurst, R., Moskal, B. M., Downey, G. L., Lucena, J., Bigley, T., & Elberb, S. "Engineering Cultures: Online versus In-class.", MERLOT Journal of Online Learning and Teaching, Vol. 4, No. 4 , pp. 438 - 445. (2008).

Classroom Experiences: Disallowing Laptops during Lectures Improves Student Performance

Leonidas Deligiannidis

Wentworth Institute of Technology
Department of Computer Science and Systems
550 Huntington Avenue
Boston, MA 02115, USA
deligiannidisl@wit.edu

Abstract - *As most Universities and Colleges are trying to increase enrollment, we often forget the problem of retention, which is a totally different problem and should be addressed separately. We often encourage our students to use their laptops during lectures for constructive work such as taking notes, design, test programs, etc. In this study I present my classroom experiences that show why, sometimes, students should not be allowed to use their laptops during lectures. Instead of using their laptops to take notes or run programs presented in a classroom, students use their laptops to engage in social activities, work on other projects or papers, etc, and as a result they do not pay attention to the material covered in class.*

Keywords: Student Retention

1 Introduction

As most Universities and Colleges are trying to increase enrollment [1-9], we often forget the problem of retention, which is a totally different problem and should be addressed separately. What do we do to keep the students enrolled in our programs? Maybe, if we focus more on retention, the problem of decreased enrollment will not be as great. Maybe we should become more effective instructions where we motivate our students and challenge them with more interesting problems during our lectures. Maybe we should pay more attention to the student evaluations and try to figure out what the problem is. Sometimes it is difficult to decipher the messages students try to get across in the course evaluations. Accreditation agencies are trying to establish a mechanism of course and self assessment. But even during these assessments, sometimes it is difficult to understand why we lose students from our programs and why many receive a failing or a low grade; which may disappoint them and drop out of a program.

Wentworth Institute of Technology is a baccalaureate degree granting institution. The Department of Computer Science and Systems has been in existence since the early nineteen eighties. Our department offers two degrees: one leading to a Bachelor of Science in Computer Science and another leading to a Bachelor of Science in Computer Networking. While the computer science degree program is a typical, traditional CS program, the computer networking program is more hands-on and exposes students to a wide skill-set in networking, computer science, and management. Four years ago, Wentworth started a program where every student receives a laptop computer for his/her studies and the cost of it and the software is incorporated in the tuition. This may seem an attractive and great idea at first. Instructors and students have the same computing environment and students can work on programming assignments anytime, not only during open lab-hours. The students can also use the computer to take notes during lectures, however, the problem irises when they use their computers for anything else but taking notes during the lectures.

We performed a study where we taught two sections of an advanced course in Network Security. In the first section, the students were allowed to use their laptops, and in the second section the students were not allowed to use their laptops. The agreement was to use the laptops to take notes. We repeated the same study using the same course and material one more semester. In this paper we present our findings and experiences between the two sets of courses; all of which were taught by the same instructor.

2 Background

It is suggested that hands-on, investigative teaching and exercises improves the learning performance of the students [7]. It is also shown that by engaging students in hands-on exercises promotes active learning and helps students develop critical thinking skills [11-12], not simply material covered in class leaving students with many unanswered questions. This improves their learning performance and as a result increases the likelihood of programs to retain their students [10-16][19]. A three year study was performed to figure out why STEM majors switch to non-STEM majors [17]. They found that students switch majors because of lack of interest, teaching methodology ineffectiveness, and because they felt overwhelmed with the curriculum demands. Successful completion of the introductory courses for the first year is crucial in retaining students in a program and most lecture courses are notoriously ineffective in engaging students [18].

3 Network Security Course

Both of our majors can choose from a pool of available Computer Science electives. One of the electives, offered to Juniors and Seniors, is our Network Security course. The book used in the class is "Cryptography and Network Security: Principles and Practice", By William Stallings, Prentice Hall. This is a great book and has been adopted by many professors offering such courses at the college level. All programming assignments are done on a laptop computer using the Java language. There are also 6 programming assignments where we use symmetric encryptions, digital signatures, digital certificates, and there are two more assignments where we implement RSA from the ground up. There is a midterm exam and a final exam as well. Because of the popularity of the topics covered in class, we offer this course during the Spring and Summer semesters every year.

4 Evaluation

For all our courses, at the end of each semester the students have an opportunity to use our Institute's faculty and course online evaluation system, where they rang and evaluated the instructor and the course. They have an opportunity to type in comments and evaluate the instructor and the course on a set of questions based on a 5-point Likert scale ranging from strongly disagree (1), to strongly agree (5). Three of the questions are used as a measure to assess how effective an instructor, which is also an overall measure for the instructors. Specifically, these three questions are:

1) The instructor stimulated thought
2) The instructor knows the subject
3) The instructor communicates subject matter well

These scores are compared against the scores of the department as well as the scores of the entire institute.
One of our early trials begun in the Summer of 2008 when I was teaching the Network Security course in two different sections. Students in Section A were not allowed to use computers in class during lectures where students in Section B were allowed to use computers during the lectures. The results are shown in Figures 1 and 2.

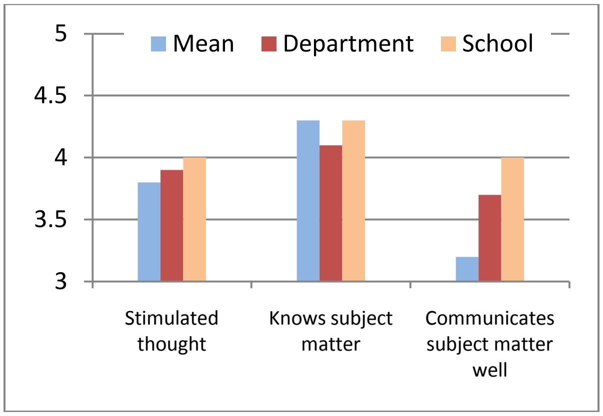


Fig. 1. Network Security (Summer 2008) Section A Computers were allowed.

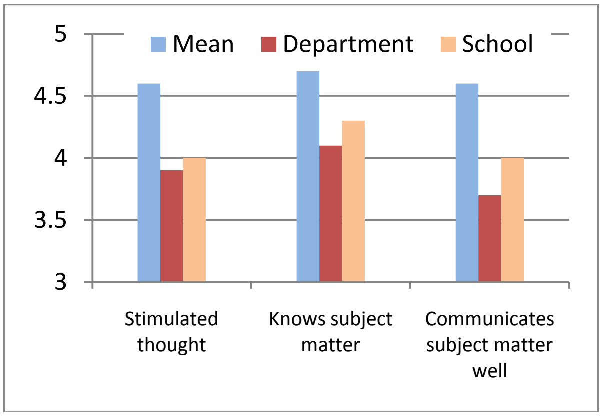


Fig. 2. Network Security (Summer 2008) Section B Computers were not allowed.

We repeated the same experiment in the following year (Summer 2009). The results were the same as shown in figures 3 and 4.

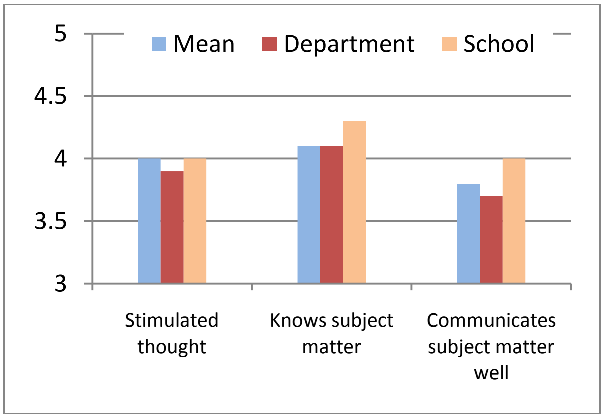


Fig. 3. Network Security (Summer 2009) Section B Computers were allowed

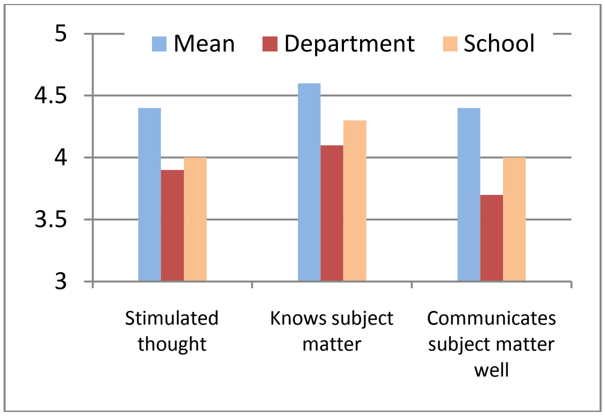


Fig. 4. Network Security (Summer 2009) Section A Computers were not allowed

During the lectures, I was paying attention at the students to explain these results. Even though the material I was covering were not amusing neither easy, many students were smiling and typing even at times that required thinking. What were they doing? They were playing games, chat with their friends, browsing the web, communicating via email, working on an upcoming assignment or paper for another class, engaging in social activities on social networks, etc.

The figures 1 thru 4 show how the students evaluated the instructor. The instructor was the same in all four classes. What changed was the perception of the students about the instructor. Students that were not allowed to use computers performed better in programming assignments and in written exams. They also thought that the course was useful, and the instructor received high scores in all three questions presented in the figures 1 - 4. So, through my experiences in teaching this particular class, I find that the students can learn more when they pay attention to the lectures and this has a direct reflection on their grades and on the way they see the instructor. Students who were using computers in class for anything other than following the lectures received low grades and even they failed the course. As a reaction, they blamed the instructor, the course, and the whole institution. They don’t understand where the problem is. It is our responsibility as educators to help them avoid such traps. By receiving low or failing grades, the students can get discouraged and disappointed and they can have their scholarships revoked. This will prevent them from continuing attending school and drop out of the school, or maybe go to a different school.

Another example that shows the same pattern is a course we offered (same instructor) in Fall 2007 and Fall 2009. The course is an introductory level course titled “Introduction to Computing and Problem Solving”. We observed exactly the same pattern, both in the course evaluation as well as in their course performance. Figures 5 and 6 show the ratings given to the instructor by the students.

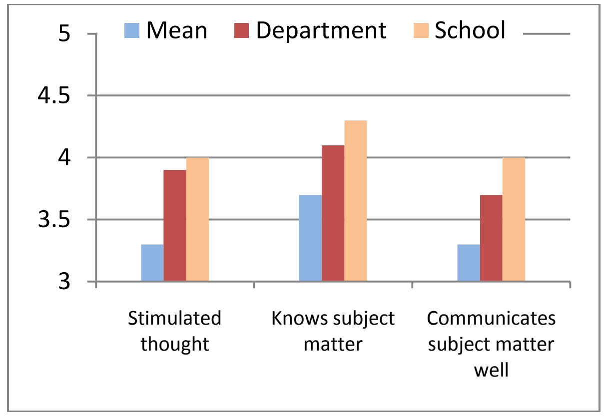


Fig. 5. Introduction to Computing and Problem Solving Fall 2007 (Computers were allowed)

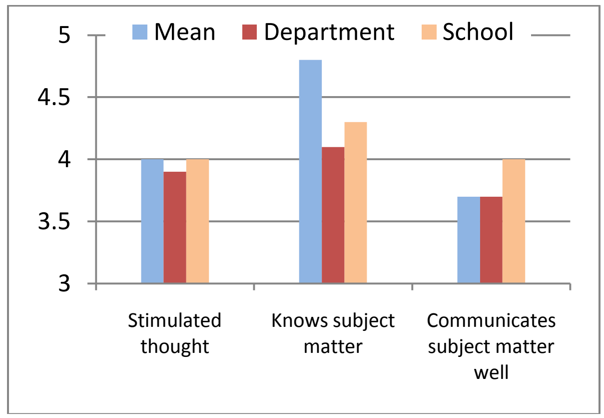


Fig. 6. Introduction to Computing and Problem Solving Fall 2009 (Computers were not allowed)

Based on these experiences, I feel confident enough to not allow the students use their laptops during lectures. The students are allowed to use their laptops in the lab during formal lab hours. Some students prefer to take notes on their laptops. I do allow such students to use their laptops after they get permission from me, which I can revoke at any point in case the usage of their computer impacts their performance by not using it to take notes.

As another example, I was teaching two sections of the Operating Systems Concepts, in the Fall of 2009. In both sections the students were not allowed to use their laptops. Not surprisingly, both sections did very well in the course and their view of the instructor was great as shown in figures 7 and 8.

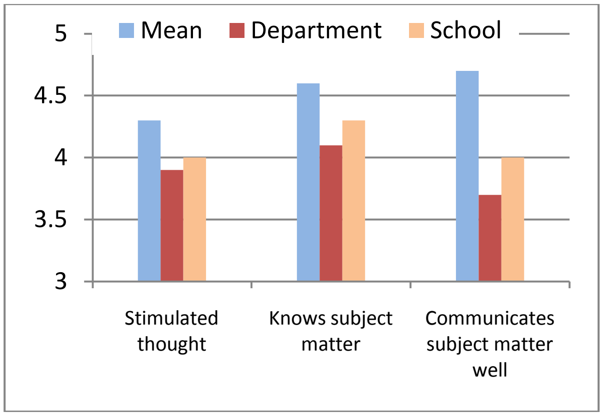


Fig. 7. Operating Systems Concepts Fall 2009 Section A (Computers were not allowed)

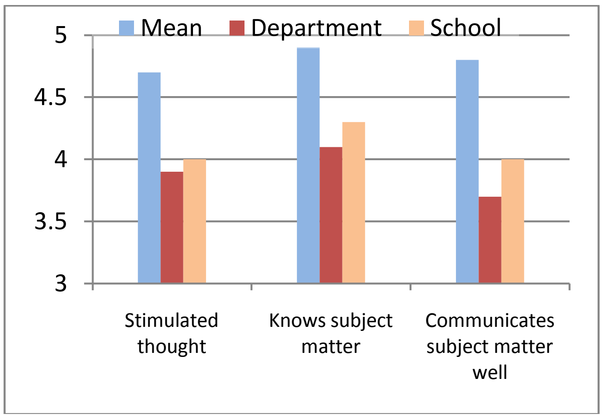


Fig. 8. Operating Systems Concepts Fall 2009 Section B (Computers were not allowed)

We observed the same pattern for the next 2 years. The results do not suggest that computers must not be allowed to all classes by all instructors. There are classes such as Network administration, and Game programming that do require the students to use their laptops during the lectures. Another variable is the instructor's style of teaching. An instructor who would ask the students to try out something that was presented just now in class would require, for example, the use of the computers.

5 Conclusions

Through my personal experiences in the classroom, I found that, in general, our students perform worse when they are allowed to use their laptops during the lectures as opposed to not being allowed. I observed that our students tend to use their laptops to check emails, engage in social activities, etc. most of the time. This activity might be acceptable at times. However, when the laptop becomes a destruction tool, the performance of the students is impacted. Low and failing grades may drive students away from a major or even from the college they are enrolled in. I only performed this preliminary study for a few semesters. I do not conclude that laptops must not be allowed in any course. However, as educators we must be more observing and alert on how the computers are being used during the lectures by our students.

6 References

[1] Anderson-Rowland Mary R. “A First Year Engineering Student Survey to Assist Recruitment and Retention”. Frontiers in Education. - Salt Lake City, Utah, 1996.

[2] Blum L. and Cortina T. J. “CS4HS: An Outreach Program for High School CS Teachers”. 38th SIGCSE Technical Symposium on Computer Science Education. ACM, 2007, pp. 19-23.

[3] Cobb Andres T. and O'Connor Terrence P. “A Model of Successful Student Recruitment for Satellite Engineering Technology Programs” Frontiers in Education. - Salt Lake City, Utah, 1996.

[4] Rafieymehr Ali Kids in Computing (K.I.C.). “Is there a Solution to Solve the Computer Science Enrollment Problem?” inroads - SIGCSE Bulletin, ACM, June 2008. - 2 : Vol. 40. - pp. 107 - 11.

[5] Vegso J. “CS Bachelors' Degree Production Grows in 2004; Poised for Decline” CRA News. - May 2005. - 3 : Vol. 17.

[6] Walker Henry M. “Advertising and Recruiting” inroads : SIGCSE Bulletin. ACM, June 2008. - 2 : Vol. 40. - pp. 16 - 17.

[7] Janet K. Yates, Madeleine Voss, Kuei-wu Tsai, “Creating Awareness about Engineering Careers: Innovative Recruitment and Retention Initiatives” 29th ASEE/IEEE Frontiers in Education Conference, Vol. 3, Nov 1999, San Juan Puerto Rico

[8] Leonidas Deligiannidis, Karina Assiter, John Russo, Michael Werner, “A Student Recruitment Strategy for Colleges and Universities", In Proc. of The 2008 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'09), pp.165-169, July 13-16 2009 Las Vegas NV, USA.

[9] Sherrene Bogle, Cynthia Onyefulu, “Assessing the predictive validity of an aptitude test as a recruitment method for undergraduate computing”, In Proc. of The 2008 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'09), pp.175-183, July 13-16 2009 Las Vegas NV, USA.

[10] Hong Wang, “From C to Python: Impact on Retention and Student Performance”, In Proc. of The 2008 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'09), pp.170-174, July 13-16 2009 Las Vegas NV, USA.

[11] John T. Tester, Jerry Hatfield; “The Design4Practice Sophomore Design Course: Adapting to a Changing Academic Environment,” ASEE Annual Conference Proceedings, June 2005.

[12] John T. Tester, David Scott, Jerry Hatfield, Rand Decker, Fonda Swimmer, “Developing Recruitment and Retention Strategies through "Design4Practice"

Curriculum Enhancements”, 34th ASEE/IEEE Frontiers in Education Conference, Oct. 2004, Savannah, GA, USA

[13] Jerry M. Hatfield, John T. Tester, “LEGO Plus”, Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition, 2005

[14] Courter, S. S.; Millar, S. B.; Syons, L. “From the students’ point of view: Experiences in a freshman engineering design course” Journal of engineering education, Vol.(87), no(3), pp 283-287, Jul 1998.

[15] Parcover, J. A., McCuen, R. H., “Discovery Approach to Teaching Engineering Design”, Journal of Professional Issues in Engineering Education and Practice, pp 236-241, Oct. 1995.

[16] Anderson-Rowland, M. R.; Blaisdell, S., Fletcher, S., Fussel, P., Jordan, C., McCarthey, M. A., Reyes, M. A., White, M., “Comprehensive programmatic approach to recruitment and retention in the college of engineering and applied sciences”, In Ptoc. Of the 29th Annual Frontiers in Education Conference, Vol.(1), 1999.

[17] Seymour, E. & Hewitt, N.M., “Talking about leaving: Why undergraduates leave the sciences”. Boulder, Co: Westview Press 1997.

[18] Carol A. Twigg, “Using Asynchronous Learning in Redesign: Reaching and Retaining the at Risk Student”, JALN Vol.(8), Issue 1, Feb. 2004 pp 7-15.

[19] Jacqueline C. Tanaka, Larry D. Gladney, “Strategies for Recruiting and Retaining Minorities in Physics and Biophysics”, Biophysical Journal Vol.(65), Jul. 1993 pp552-558

Experimentation with Tutors for Teaching Mathematical Reasoning and Specification

S. Drachova-Strang[*], M. Sitaraman[*], and J. Hollingsworth[**]
School of Computing, Clemson University[*], Clemson, SC, USA
Department of Computer Science, Indiana University Southeast[**], New Albany, IN, USA

Abstract - *As software is finding increased use in mission and life-critical applications, its correctness is even more important. The shift towards verified software places new skill requirements on the next generation of software engineers. To produce high quality software, the future developers need the ability to reason about components mathematically, and to understand and write formal specifications using precise language of mathematical notation. This paper discusses online tutorial modules developed to teach a variety of related mathematical skills to students of computer science. The modules are based on a set of learning outcomes, which in turn are based on a set of basic reasoning principles called the Reasoning Concept Inventory. The modules can be adapted for use at different levels of the computer science curriculum. Preliminary experimentation conducted in a controlled classroom setting has produced positive results.*

Keywords: Contracts, formal specification, learning outcomes, mathematical reasoning, tutorials.

1 Introduction

Developing software according to formal contracts has great implications in all areas of software engineering. Formally specified and verified software is more reliable, experiences fewer failures, and costs less time-wise and effort-wise at the maintenance stage. Given the last decade's shift towards modular development and reusability, it is especially critical to design components that adhere to formal specifications [1, 2]. The concept of "design-by-contract", coined by Bertram Meyer [3] in relation to his design of Eiffel programming language, explains how software components should collaborate. Some programming languages, such as Eiffel [4] and RESOLVE [5], have built-in support for specifications, while others have to rely on a separate specification language, such as Java on JML [6] and C# on Spec# [7].

In the modern component-based software development paradigm, software components are developed, modified, and maintained by different engineers separated from each other geographically and/or chronologically, and the central mechanism to guarantee correctness of their design and seamless interoperability is development of verified software components according to formal specifications. To design software according to formal specifications, software engineers need to be able to understand mathematical notations, employ their mathematical reasoning skills to reason about components and their relationships, and be able to write mathematical assertions. Elsewhere, we have discussed a detailed conceptual inventory of skills that software engineering students need to possess [8]. As indicated in the study, specific mathematical reasoning skills are of particular importance. However, in a typical computer science curriculum there is no well-known paradigm of teaching such skills. In addition, software engineering students often have diverse mathematical backgrounds and skills.

Our experience has shown that the students themselves are receptive to the need for formalism, and are interested in learning mathematical reasoning skills in order to be able to write correct code. To educate these future practitioners effectively, we are proposing a tutoring system to aid in teaching mathematical aspects of specification and reasoning as described in detail in the following sections.

In [8], a set of six basic reasoning principles for computer science students have been identified as being essential to the development of high quality software. These six principles are referred to as the reasoning concept inventory (RCI). In upcoming sections of this paper we discuss three different tutorials that directly relate to two of the basic reasoning principles found in the RCI. These two reasoning principles are:

RCI#2 "In order to model software components, one needs familiarity with basic discrete math structures, such as sets, strings, integers and other number systems, relations, and functions."

RCI#3 "Precise (mathematical) specifications for software components are critical in order to reason about component-based software and establish its correctness."

For RCI#2 and RCI#3 we have developed specific *learning outcomes*, which we have used to help organize our tutorials. That is, our tutorials have been specifically developed to help students achieve the learning outcomes that are based on the RCI principles essential for effectively reasoning about the correctness of software.

The learning goals and specific learning outcomes are also used in multiple ways. First, because we have these specific learning outcomes, we can use them to help us develop our instructional materials, including tutorials (as will be seen in upcoming sections). Second, our formative assessments (based on these specific learning outcomes) can help us make "mid-course corrections" when we see results that show lower than expected student learning during the semester. Furthermore, our summative assessments (at the end of the semester) give us feedback that we can use for future process improvement, i.e., improved instructional materials, adjustments in how much time is spent on a particular concept, etc. We can also use these learning goals and outcomes as a tool for curriculum alignment, i.e., making sure our undergraduate computer science courses collectively cover important reasoning-related topics.

2 Background

2.1 Related work

As it has been previously shown [9][10][11][12], tutorials significantly improve student performance in many areas of the computer science. While there is a plethora of tutorials to teach a variety of programming, there are few tutorials for teaching students mathematical reasoning skills and the correct use of mathematical notation that will enable them to reuse and specify software components leading to the production of verifiable software.

While deciding on the type of the tutorial that would satisfy our needs we have reviewed a number of available studies and their results.

We summarize three tutoring systems that are most related to our work here. The Syrus system, developed by researchers at the Ohio State University and described in [12], is a tutor that teaches mathematical skills and predicate logic in particular. The tutorial system is capable of generating a large number of problems, and provides a student with immediate feedback and explanation. Kumar's Problets [13] are useful for teaching a variety of programming skills across several programming languages; they provide instant feedback, code walk-through, scoring, and special tools for the instructors. The tutorial in Leonard et al. [14] teaches mathematical thinking and helps students understand assertions by utilizing an execution-based approach with rapid feedback. It has been shown in a past study that immediate feedback helps students retain correct information [15].

2.2 General features of the Tutorial System

Our system is an interactive online tutorial with JIT ("just-in-time") feedback. It teaches basic mathematical concepts and notation, and a language for specification. Each topic is presented using a number of short, easy to read slides with a simple interface (shown in Figure 1), followed by practice questions. Each section contains 5-10 true/false or multiple-choice problems at the end. When a question is answered, the tutor dynamically checks the correctness of the student's answer and provides a timely detailed feedback.

Our tutorial is used to supplement classroom instruction at the undergraduate level, and has been used in a junior-level software engineering course at Clemson. To teach formal specification and reasoning principles, we use RESOLVE as a medium of choice [5]. RESOLVE is a unique object-based programming language that includes a built-in specification language. Its development environment includes a compiler, a verifier [16], and now this tutorial.

Formal specification languages, such as RESOLVE, use mathematical models to conceptualize programming objects, for example, mathematical strings or sequences for programming stacks or queues. A component's operations are specified via pre- and post-conditions using suitable notations from the corresponding mathematical models. The specification of *Push* operation on Stack, for example, is expressed using string theory notations. At this time, the tutorial contains three well-developed modules: mathematical string theory, specification language, and specification-based test-case generation. The tutorial can be adapted to any language of interest, since all specification languages will require knowledge of similar concepts.

3 Tutorial module I: String Theory

3.1 Mathematical String Theory

This module introduces general mathematical notation along with the basic notations of the mathematical string theory, such as empty string, the length of the string, string concatenation, and string reversal.

A mathematical string is a sequence of alphanumeric characters and/or special symbols. Strings are defined over an alphabet set. If Sigma (Σ) is an alphabet, then all the strings over that alphabet are defined as Σ*. Strings are usually denoted by using the beginning letters of the Greek alphabet: α (alpha), β (beta), γ (gamma), δ (delta), etc. Variables are usually named using the end of the Latin alphabet: *x, y, z,* etc. To indicate the variable type we use colon. To indicate the value of the string we use angular brackets. For example, if we have a string of integers α, then we can write: α = <5>.

An empty string is represented by the Greek Lambda (Λ). The binary infix concatenation operator requires two operands with the concatenation symbol positioned between them. (Different systems use a variety of symbols to represent concatenation). In our tutorial we denote concatenation by a lower-case letter "o". So <1> o <4, 4, 5> = <1, 4, 4, 5>. The string length (unary) operator is denoted by two vertical bars | |, e.g., |<1, 2, 5>| = 3.

Some interesting properties of strings are also discussed in the tutorial; students need to understand them for later exercises in proving correctness. Some properties of an empty string are described below:

1. *The length of an empty string is zero:* $|\Lambda| = 0$;

2. *Any non-empty string concatenated with an empty string produces the same non-empty string:* $\Lambda \ o \ \beta = \beta$;

3. *Reversal of an empty string is an empty string:* *Reverse* $(\Lambda) = \Lambda$;

The 10 practice exercises start with the easy ones that test the understanding of each basic concept, and then progresses to more difficult ones that combine all the introduced concepts to see how students have generalized and transferred the material. For example, a simple exercise will ask the student to decide about the length of a given string, while a more difficult exercise would be a selection of true/false in a case that combines string reversal and concatenation:

Reverse ((α o β) o Λ) = Reverse (β) o Reverse (α);

Figure 1 shows the screen shot of a tutorial page and Figure 2 shows an exercise along with a student's answer and feedback. Understanding these simple theorems is very important, as it helps students prove verification conditions of correctness for a piece of software modeled using mathematical string theory, and thus see the connection between the mathematics they learn and software correctness.

3.2 Experimental settings and results

The experiment was conducted in a software engineering classroom with 18 undergraduate students randomly divided into groups of an approximately equal size. The test group took the tutorial and went over the practice questions where they received detailed feedback, while the control group did not take the tutorial. Afterward, a test was given to both groups to check the understanding of the topic.

The results are summarized in Table 1. They indicate that students in the test group performed better on the test after taking the tutorial. The average score of the control group is 6.2 out of 8 indicates 77.5% success, while the students in the test group had an average of 7.75 out of 8, i.e. 96.8% success. An anecdotal record indicates that several students mentioned that they were glad they took the tutorial, because it explained a few things they did not quite understand before that.

Table 1. Results of the String Theory Tutorial

	No. of Students	Average
Control group	8	77.5%
Test group	9	96.8%

3.3 Learning Outcomes for this module

Reasoning concept inventory item number two (RCI#2 – defined in Section 1) has to do with the discrete structures that are often taught in computer science discrete math classes. The general learning goal we have developed for RCI#2 is: *Students will use mathematical discrete structures to organize and model a software system's data*. There are a multitude of specific learning outcomes that are related to this general learning goal. The learning outcome most related to the tutorial discussed in this section is: *Students will calculate the result of applying string operators to specific mathematical strings*. As you can see from Figures 1 and 2 in this section, we have developed both instructional material and formative assessments relating directly to the specific learning outcome stated above, where the string operators include concatenation and reversal of strings of items.

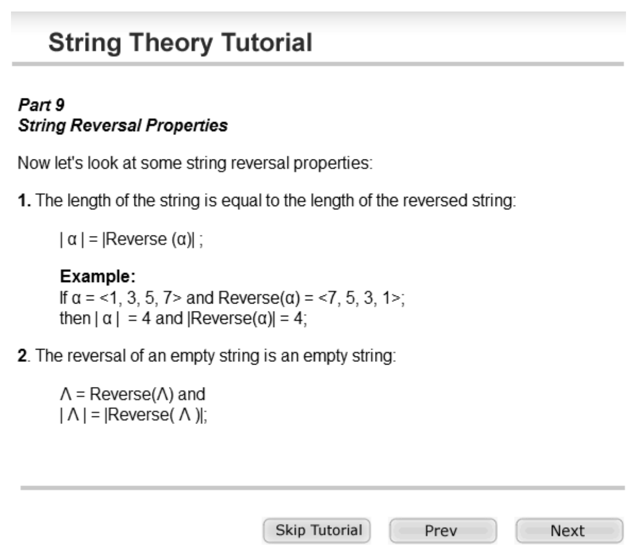


Figure 1. A string theory tutorial page

4 Tutorial module II: Specifications

4.1 Basics, equivalence, and redundancy

RESOLVE uses mathematical modeling to specify data structures and their behavior. Operation specification is one

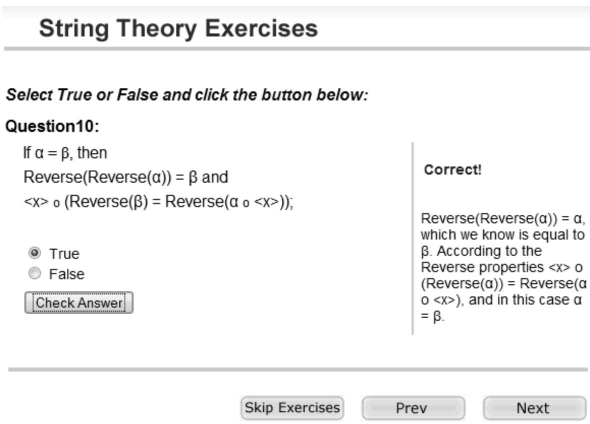


Figure 2. A string theory tutorial exercise.

of its specification mechanisms. The three operation specification components are: *operation signature* that specifies names, types, and parameter modes for all the formal operation parameters; *requires clause* - a precondition that should be true before an operation is called; and *ensures clause* - a post-condition that will hold after the operation executes. All three components are illustrated in the following operation header:

Operation Increment (**updates** Num: Integer);
requires: Num < Max_Int;
ensures: Num = #Num + 1;

This tutorial explores operation parameter modes and explains redundant and equivalent specifications. Parameter modes explain what happens to the value of the incoming variable after the operation completes. To illustrate the connection between mathematical string notations and software specifications, we consider the specification of stack behavior where a Stack object is modeled mathematically as a string. Given this model for a Stack, we provide the specification of *Push* (as an example), which can be written as shown below:

Operation Push(**alters** E: Entry; **updates** S: Stack);
requires: |S| < Max_Depth;
ensures: S = <#E> o #S;

In the *requires* clause, Max_Depth is the max number of elements allowed on the (bounded) stack. In the *ensures* clause, the pound sign (#) means the incoming value of the variable. The outgoing value (new) stack is now the new element concatenated with the contents of the initial stack. *Equivalent* specifications can be written in more than one way, but have the same meaning, such as shown below:

Operation Push(**alters** E: Entry; **updates** S: Stack);
requires: |S| + 1 <= Max_Depth;

The *ensures* clause of Push expressed below is an example of a *redundant* specification:

ensures: S = <#E> o #S and |S| = |#S| + 1;

The second conjunct implies that the length of the new stack will now be longer by one element, so the conjunct "*|S| = |#S| + 1*" is redundant. Redundant specifications should be avoided because readers might be confused by their presence and might wrongly believe that they misunderstand the specifications. Similar to the tutorial on mathematical string theory, this part of the tutorial is followed by a number of exercises (see Figure 3).

4.2 Experimental settings and results

This experiment was conducted in similar settings. Students were randomly divided into the test group that took the tutorial, and the control group that did not. The average score of the control group is 9 out of 10, which indicates 90% success, while the students in the test group had an average of 9.5 out of 10, i.e., 95% success. A detailed summary is presented in Table 2. Apparently, students understood the ideas well enough from the lectures that the tutorial was of marginal help in this case.

Table 2. Results of the Parameter Mode Tutorial

	No. of Students	Average
Control group	11	90%
Test group	7	95%

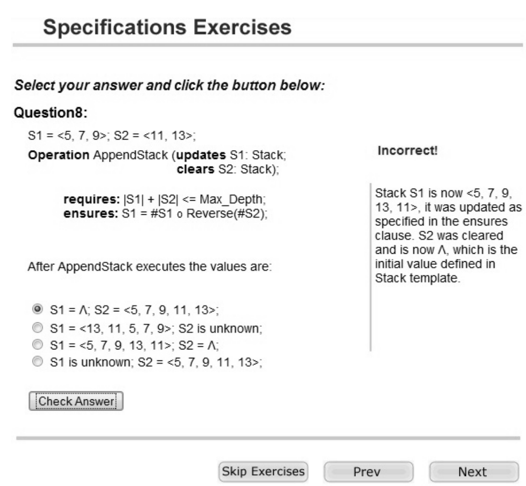


Figure 3. An exercise from specification tutorial

4.3 Learning Outcomes for the specification basics module

The tutorials in this and the following section pertain to RCI#3 (as defined in Section 1) and the general learning goal we have developed for RCI#3 is: *Students will use a component's mathematical specification to be able to reason effectively about a component's behavior*. In Figure 3, the student is asked to use the formal specification for an operation to determine the correct output from substituting that operation's specification with a particular set of inputs. This part of the tutorial supports the general learning goal listed above, and the following specific learning outcome: *Students will determine analytically the results expected from calling an operation by using the operation's specifications and instructor provided inputs.*

5 Tutorial Module III: Understanding specifications using test cases

5.1 Understanding test cases

To assure the quality of a software component, it is necessary to show that it is built according to the specifications. One of the advantages of writing formal specifications for an operation is the possibility of checking its correctness by examining its pre- and post-conditions. A set of valid inputs is determined by the requires clause of an operation, and a set of the expected outputs is determined by the ensures clause of an operation. These pairs of inputs-outputs are the specification-based test cases for an operation. If either input or output violates the specification, the pair is invalid. If a student understands the specification he/she can determine which test case is valid.

This tutorial teaches students to determine valid test cases for a variety of operations. This part of the tutorial has several exercises with multiple-choice answers. Each of them is rather complex and requires students to spend some time studying specifications to decide which test cases are valid/invalid. There is usually more than one correct test case. We use "non-descriptive" operation names to ensure that students understand the specification instead of deducing what the operation does from its name.

5.2 Experimental settings and results

This experiment was conducted in a different way to study the impact of tutorials on students who had trouble understanding a concept. So first an initial quiz was given to all students, without any tutorials. Then a second quiz (similar to the first one) was given. Students who performed poorly on the first quiz were given the tutorials. The summary is presented in Table 3.

Table 3. Results of the Test Case Generation Tutorial

	Baseline quiz avg.	Second quiz avg.
Control group	79%	92%
Test group	59%	93%

Though the test group scored only a slightly better average on the test, the tutorial has helped more than it appears. This becomes clear when we consider the results of the baseline quiz that students took during the previous class session, and the students now in the tutorial test group had previously scored an average of 5.9 out of 10 points on that quiz, while the students now in the control group had a previous average of 8.8 out of 10. So, if the control group students have shown a 16% improvement over the previous quiz, the improvement for the test group was more than three-fold at 58%. The results indicate that our tutorial is in fact an effective learning tool especially for students who have difficulty in mastering a certain concept.

A related earlier study using a test case reasoning assistant (TCRA) tool also had been conducted at Clemson University [14]. The tool uses a simple interface to walk the user through a set of test case creation exercises with rapid feedback and data collection. The test group students averaged 94% after the use of the TCRA tool compared to the control group average of 78%. The data from both our study and theirs [14] indicates that students do in fact benefit from use of specification learning tools with instant feedback.

5.3 Learning Outcomes for this module

The tutorial in this section also pertains to RCI#3 and has the specific learning outcome: *Students will demonstrate their understanding of an operation's specifications by evaluating an operation on specific instructor provided test points*. As can be seen in Figure 4, a student is asked to select which test case (valid inputs and expected outputs) is correct with respect to a formal specification of an operation. Again, this is a formative assessment that can be used by both the student and the instructor to see how well the student understands the material.

6 Discussion and future work

The results of the experiments are promising, and have demonstrated on a small scale that tutorials are useful in teaching mathematical aspects. Understanding of the basic concepts of mathematical string theory and notation, specifications, parameter modes, and specification-based software testing techniques are important to mastering the idea of formal contracts. These topics are also an important supplement to the undergraduate software engineering course curriculum. In the future, we intend to develop another level of tutorial for competent programmers.

As was noted in Section 3.3, there is a multitude of specific learning outcomes related to the general learning goal of teaching students how to use discrete data structures to model a software system's data. So, one way to expand the functionality of our tutorial modules is to increase coverage of the learning outcomes corresponding to the

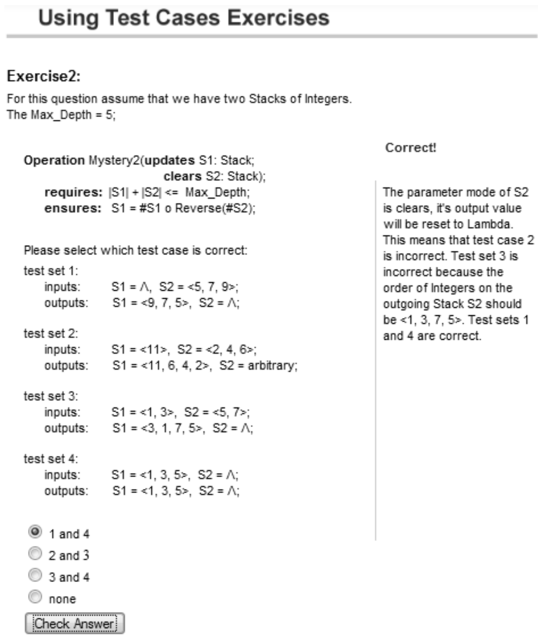


Figure 4. A specification-based test case tutorial exercise

principles in the Reasoning Concept Inventory.

There are other ways to expand the functionality of the tutorial modules. For example, we are working on introducing automation for generating exercises from various mathematical theory files. Similarly, we are planning to generate specification and test case exercises automatically. This will provide virtually an unlimited number of such exercises that can be produced on demand from several built-in templates. Each exercise will send the student's answers to the verifier to determine their correctness and then provide feedback. Taken together, all these modules will help students to learn the mathematical aspects of computer science and software engineering more effectively.

7 Acknowledgements

This research is funded in part by NSF grants DUE-1022941 and DUE-1022191. The authors of this paper would like to thank all the participants of the experiment and all the members of the Clemson RSRG for the group discussions on the topic.

8 References

[1] Larson, P.G., Fitzgerald, J., Brooks, T. Applying Formal Specification in Industry. *IEEE Software*, 13, 3, (May 1996), 48-56.

[2] Fraser, M.D, Kumar, K., Vaishnavi, V.K. Strategies for incorporating formal specifications in software development. *Communications of the ACM*, 37, 10, (Oct. 1994), 74-86.

[3] Meyer, B. Applying "Design by Contract". *Computer (IEEE)*, 25, 10, (Oct. 1992), 40-51.

[4] Meyer, B. *Eiffel: The Language.* Prentice Hall, Englewood Cliffs, N.Y., 1992.

[5] Edwards, S.H., Heym, W.D., Long, T.J., Sitaraman, M., and Weide, B.W. Specifying components in RESOLVE. *Software Engineering Notes* 19(4) (1994), 29-39.

[6] Leavens, G. T., Baker, A. L., and Ruby, C. JML: A Notation for Detailed Design. In Haim Kilov, Bernhard Rumpe, and Ian Simmonds (editors), *Behavioral Specifications of Businesses and Systems*, ch. 12, (1999), 175-188.

[7] Barnett, M., Leino, K.R.M., Schulte, W. "The Spec# Programming System: An Overview." *Proceedings of Construction and Analysis of Safe, Secure and Interoperable Smart Devices (CASSIS)*, Marseilles. Springer-Verlag, 2004.

[8] Krone, J., Hollingsworth, J.E., Sitaraman, M., and Hallstrom, J. A. *Reasoning Concept Inventory for Computer Science*. Technical Report RSRG-09-01, Clemson University School of Computing, September 2010.

[9] Kumar, A. Results from the Evaluation of the Effectiveness of an online tutor on expression evaluation. *SIGCSE* 36, (Feb. 2005), 216-220.

[10] Sykes, E. R, Franek, F.: An intelligent tutoring system prototype for learning to program Java. *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT '03),* Athens, Greece, (July 2003), 485-486.

[11] Del Vade Virseda, R., Fernandez P., Munos, S., Murillo, A.: Intelligent Tutoring System for Interactive Learning of Data Structures. In: *ICCS* 2009, Part II, LNCS 5545, (2009), 53-62.

[12] Syrus tutoring system: http://syrus.cse.ohio-state.edu/overview.html

[13] Kumar, A., Generation of Problems, Answers, Grade, and Feedback – Case Study of a Fully Automated Tutor. *Journal on Educational Resources in Computing (JERIC).* (2005), 5, 3. DOI =http://doi.acm.org/10.1145/1163405.1163408

[14] Leonard, D.P., Hallstrom, J.O., Sitaraman, M. Injecting Rapid Feedback and Collaborative Reasoning In Teaching Specifications. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, ACM. (2009) 524-528.

[15] Gee, J. P. Learning By Design: Good Video games as Learning Machines. *E-Learning and Digital Media*, (2005) 2, 1.

[16] Sitaraman M., Adcock B., Avigad J., Bronish D., Bucci P., Frazier D., Friedman H., Harton H., Heym W., Kirschenbaum J., Krone J., Smith H., Weide B.W.: Building a Pushbutton RESOLVE Verifier: Progress and Challenges. To appear in *Formal Aspects of Computing*, Springer (2010)

Approaching quality standards to e-learning

Daniel Pons, José R. Hilera, and Carmen Pagés
Department of Computer Science, University of Alcalá, Alcalá de Henares, Madrid, Spain

Abstract - *Quality applied to e-learning should be achieved thanks to the implementation of a quality standard in the structural organization which provides an on-line education system. A generic quality standard could be adopted, but specific quality standards focused on e-learning have been developed. The internal similarities and differences that can be found among quality standards applied to e-learning are shown in this study.*

Keywords: E-learning, quality, standards.

1 Introduction

A quality educative experience is aimed in e-learning, but the quality concept is widespread and diffuse, and there are several methods for facing a quality improvement. Thus, many interpretations and perspectives about what e-learning quality might mean have appeared [1]. At this juncture, the initial starting point for an organization to improve its internal performance is adopting quality standards. There exists generic quality standards which can be adopted in e-learning and education[2], but there have also appeared specific e-learning quality standards.

In this paper are presented the results of an analysis and comparison process among a set of quality standards. It is intended to study and assess the applicability to e-learning of different quality standards. In a first step, a descriptive summary of each standard is shown. Secondly, a standard classification in terms of their approach is given. Finally, the result of a match data crossing among the identified data elements in the standards is shown for obtaining information on the internal similarity among the standards.

2 Quality standards

Taking as a starting point the classification system of e-learning standards developed by Hilera and Hoya [3], based on twelve categories that group over a hundred existing standards identified by the LTSO initiative promoted by CEN [4], there have been selected the standards grouped in the quality section to perform this study. These quality approaches are:

- CWA 14644 [5] provides a descriptive framework to classify quality standards. CWA 14644 is focused on establishing a standard categorization for process-oriented standards, offering descriptive criteria for quality standards. It can be considered as a first step to define metadata on quality standards, since its purpose is to catalogue standards according to its generic information. The classification process is established in two main groups:

  1. Process-oriented standards.

  2. Product-oriented standards.

- CWA 15533 [6] includes a metadata model, richer than the model supplied in CWA 14644, to describe quality approaches.

- CWA 15660 [7] shows specific examples of implementing quality management in e-learning, some of them using the ISO/IEC 19796.

- CWA 15661 [8] is focused on the life cycle of e-learning, and is structured around a set of integrated blocks to help in the selection process of an educational resource. For this purpose it specifies a data element profile called LST or Learning Supplies Transparency, which can be put on a level with IEEE LOM metadata schema. Therefore metadata schemas can be useful for automatic discovery of learning resources.

- EFQM Excellence model [9] and UNIQUE [10] as generic quality approaches.

- The Spanish UNE 66181 [11] standard provides guidelines for the identification of characteristics that define the quality of virtual education.

- ISO/IEC 19796, which is the first specific e-learning quality approach. Two parts of this standard are relevant for this paper, ISO/IEC 19796-1 [12] defines a framework of implementing quality, and ISO/IEC 19796-3 [13] includes methods and metrics related to quality.

3 Clasifying quality approaches

We define three models of approach, based on the CWA 14644, and incorporate a typology for standards that provide information of the learning resource:

1. Descriptive models to analyze quality proposals.

2. Process models of quality management.

3. Product models.

The first approach is related to standards that attempt to define concepts used to classify quality standards. These concepts can be assumed as quality metadata. In the second and third cases, proposals will be a specific quality approach. The second model includes process-oriented approaches, i.e., quality standards to be applied to the life cycle of e-learning ranging from the development planning to the completion of product use. The third approach contains product oriented standards used to determine the quality of the educational process by providing information of the functional model, such as information related to specific educational resource, student, technical, administrative or educational aspects.

These three models lay down a first categorical differentiation to classify quality standards, as shown in Table 1.

Table 1. Models of quality approach.

	CWA 14644	CWA 15533	CWA 15660	CWA 15661	EFQM	UNIQUE	ISO/IEC 19796	UNE 66181
Descriptive models	X	X					X	
Process models		X	X		X	X	X	
Product Models				X				X

4 Analysis of quality standards

There exists concepts and data elements in the quality approaches that could be considered peers. Wirth [14] stated that different standards, though a priori might seem different, represent similar information. The aim of the current analysis is to determine the coverage, completeness and applicability of quality standards, and find the similarities and differences in methods and data models proposed by quality standards taking into account the different nomenclatures and levels of detail concerning the representation of information. The following will show the results of the study.

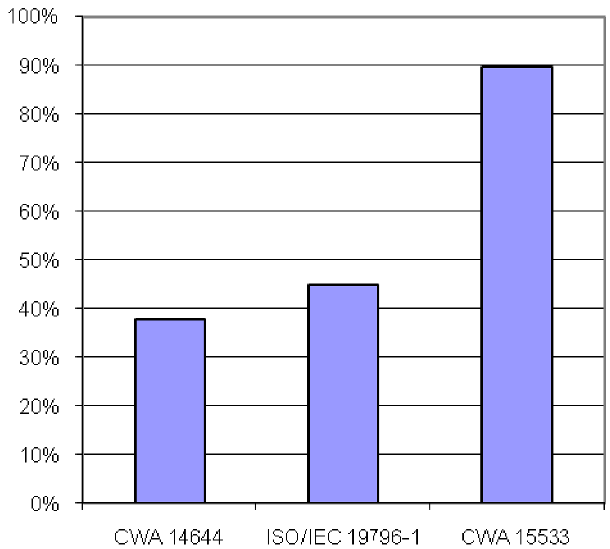
4.1 Metadata quality standards

CWA 14644 is a first pattern for a classification of quality standards; hence its concepts are scarce compared with other standards. The conceptual model proposed by the CWA 15533 standard enlarges upon the data element set and provides a greater level of detail, meaning a step in the standardization of metadata for quality approaches in e-learning.

The ISO/IEC 19796-1 standard is a process-oriented approach to implement a quality management. It provides a framework suitable for e-learning, from which it has been extracted relevant information to implement a metadata approach for the classification of quality standards.

Understanding the descriptive model as a metadata system of quality standards, the Figure 1 compares statistically the degree of information provided by these metadata quality standards. CWA 15533 is the most comprehensive in coverage of data elements and information provided for the classification of quality standards.

Figure 1. Statistics quality standards focused on the descriptive model.



4.2 Process for implementing quality

In order to compare quality management processes it is helpful to previously establish a basis that allows to link different concepts: the process flow of implementing a quality system involves a cycle of chained events, which may be linear or cyclical, but in order to reduce the complexity of linking methods, processes and workflows will not be taken into account.

The applied methodology can be summed up in a few steps: In the first instance, a study of the stages and concepts of each approach has been done. Secondly, the main categories of each standard have been extracted to establish generic categories which will group similar or homogeneous concepts related to quality approaches. In this stage, ISO/IEC 19796-1 shows a wide, linear structure, mainly used for the creation of the categories. Afterwards, a crossing of information among the different standards has been done. Finally, a summary of the results is reflected in Table 2, which indicates the amount of information coverage provided for the different standards according to the legend. The Total Weight row can be used as estimated global value of the breath of scope for each standard.

Table 2 legend:

- VL: Very low coverage, one data element (weight=1).
- L: Low coverage, two data elements (weight=2).
- M: Medium coverage, three data elements (weight=3).
- H: High coverage, four data elements (weight=4).
- VH: Very high coverage, five or more data elements (weight=5).

Table 2. Standards coverage of quality process for e-learning.

Categories	**ISO/IEC 19796-1**	**CWA 15660**	**CWA 15533**	**EFQM**	**UNIQUE**
Application					M
Needs analysis	VH	M	M	L	VH
Context	VH	L	L	M	M
Design	VH	H	VH	VL	M
Development	VH	H	M	VL	VL
Learning Process	M	VL	M	L	L
Evaluation	H	M	VL	H	L
Total Weight	27	17	17	13	19

ISO/IEC 19796 is identified as the most representative standard thanks to its completeness in terms of concepts taken into account for being used in an implementation of quality in e-learning. The key to the success is a complete, consistent life cycle definition. It must be emphasized that this standard involves quality in each step, as each stage has a description of the process in a quality framework based on standards, metrics, objectives and results.

However, ISO/IEC 19796 does not cover all the concepts associated with the term “quality” in all its meaning. As an example, UNIQUE approach is the only one which includes the processing phase of an application for certification within its data model.

On the other hand, the CWA 15660 standard is not a quality standard for e-learning itself, but it defines good practice for quality, and hence it is possible to extract steps to achieve success in implementing the quality.

4.3 Educational resource information

CWA 15661 and UNE 66181 contain information regarding learning resources. It has therefore been included in this group the IEEE LOM standard [15], which is not a quality standard but an educational resource metadata standard.

A complete cross of data elements among CWA 15661, UNE 66181 and IEEE LOM has been done. CWA 15661 is taken as principal of this group for defining categories because of its broad definition of data in terms of providing an educational resource with useful information available, hence the total number of items practically matches with the items contained in the CWA 15661 standard. It was observed that these standards include internal relations and similarities in their elements.

Table 3 is a summary of the entire process of analysis where each data element has been crossed and matched. It contains the identification of the number of data elements that the standards include for each category, and allow us to have an idea of the global coverage of each standard.

Table 3. Number of data elements in quality standards and learning objects metadata.

Categories	**Total identified items**	**Items in CWA 15661**	**Items in UNE 66181**	**Items in IEEE LOM**
Provider information	11	11	1	6
Overview of the educational resource	13	12	5	9
General organization of the educational resource	10	10	9	3
Technology	6	6	5	3
Registration information and management	3	3	1	2
Resources or digital content used	8	8	5	3
Learning face to face	6	6	3	1

or virtual classroom events				
Collaborative learning and interaction	5	5	5	0
Student support	7	7	5	1
Student assessment	11	11	0	0

UNE 66181 is the only standard which contains a real system to assess the quality of e-learning, while CWA 15661 and IEEE LOM include common descriptive metadata involved in the process of educational action. UNE 66181 contains reference to these concepts or metadata, but taking them from a more pragmatic view thanks to implementing a system of assessment and scoring of each item to establish selection criteria of e-learning resources. It should be noted that UNE 66181 provides the concept of "market demand" to be taken into account in the evaluation system of e-learning resources.

5 Conclusions

As a first result of this study, the existence of a variety of standards for e-learning is evidenced. Some of them have very similar contents, since they are based on each other. The existence of redundancies in the quality standards for e-learning is detailed. That is to say, standards include equivalent concepts represented by different data elements or concepts, and the amount of similarity is detailed and measured with overall weights.

Another result obtained in the light of the research is the fact that common grounds exists among CWA 15661, UNE 66181 and IEEE LOM. It makes it possible to conclude that learning objects metadata store relevant information to the implementation of a process of quality management.

It is noteworthy to highlight ISO/IEC 19796, the latest from the analyzed standards and with some of its parts still under development, thanks to the completeness and accuracy of this standard in terms of concern about the process of implementing quality in an institution or organization dedicated to online training providing.

Standardization institutions are constantly undertaking a review of their standards and developing new ones so as to produce a restructuring of the current scene, becoming a state of depreciation of those standards which have been rewritten. Due to the diverse range of available standards as well as redundant information contained in them, harmonization of standards for e-learning is a task that, although there are some technical hitches difficult to circumvent owing to the number of existing standards, will greatly ease the complexity of adopting quality standards.

6 Acknowledgements

This research is partially funded by the University of Alcalá (grant UAH/EV381).

7 References

[1] Ehlers U, Pawlowski JM. “Quality in European e-learning: An introduction”; In: Ehlers U, Pawlowski JM. Handbook on Quality and Standardisation in E-Learning. Essen, Germany: Springer, 2006.

[2] Pawlowski JM. “Adopting quality standards for education and e-learning”; In: Ehlers U, Pawlowski JM. Handbook on Quality and Standardisation in E-Learning. Essen, Germany: Springer, 2006.

[3] Hilera JR, Hoya R. “Estándares de e-learning: Guía de consulta”. Alcalá de Henares: University of Alcalá; 2010 [cited 2011 Mar 01]. Available from: http://www.cc.uah.es/hilera/GuiaEstandares.pdf

[4] CEN. Learning Technologies Standards Observatory [homepage on the Internet]. 2011 [cited 2011 Mar 01]. Available form: http://www.cen-ltso.net.

[5] European Committee for Standardization. CWA 14644:2003. “Quality Assurance Standards”. CEN, 2003.

[6] European Committee for Standardization. CWA 15533:2006. “A model for the classification of quality approaches in eLearning”. CEN, 2006.

[7] European Committee for Standardization. CWA 15660:2007. “Providing good practice for e-learning quality approaches”. CEN, 2007.

[8] European Committee for Standardization. CWA 15661:2007. “Providing e-learning supplies transparency profiles”. CEN, 2007.

[9] European Foundation for Quality Management. EFQM Excellence Model [homepage on the Internet]. 2011 [cited 2011 Mar 01]. Available from: http://www.efqm.org.

[10] European Foundation for Quality in E-Learning. UNIQUe European University Quality in eLearning. [homepage on the Internet]. 2011 [cited 2011 Mar 01]. Available from: http://www.qualityfoundation.org/unique-certification/.

[11] AENOR. UNE 66181. “Gestión de la calidad. Calidad de la formación virtual”. AENOR, 2008.

[12] International Standards Organization. ISO/IEC 19796-1. “Information technology – Learning, education and training – Quality Management, assurance and metrics. Part 1: General approach”. ISO, 2005.

[13] International Standards Organization. ISO/IEC 19796-3. “Information technology – Learning, education and training – Quality Management, assurance and metrics. Part 3: Reference methods and metrics”. ISO, 2009.

[14] Wirth MA. “An analysis of international quality management approaches in e-learning: Different paths, similar pursuits”. In: Ehlers U, Pawlowski JM. Handbook on Quality and Standardisation in E-Learning. Essen, Germany: Springer, 2006.

[15] Institute of Electrical and Electronics Engineers. IEEE 1484.12.1-2002. “Learning Object Metadata". IEEE, 2002.

The use of computers for graduate education in Project Management. Improving the integration to the industry.

Ordieres-Meré J[1].; González-Marcos A[2].; Alba-Elías F[2].
[1]ETSII. Technical University of Madrid (UPM). c/ José Gutiérrez Abascal 2, 28006 MADRID; SPAIN
Visiting Professor at the Mechanical and Industrial Department of the University of Iowa. IA. USA
Email: j.ordieres@upm.es.
[2]Department of Mechanical Engineering. University of La Rioja (UR). c/ Luis de Ulloa 20, 26004 LOGROÑO; La Rioja. SPAIN

Abstract - *This paper presents an initiative for monitoring the competence acquisition by a team of students with different backgrounds facing the experience of being working by projects and in a project. These students are graduated bachelor engineering are inexperienced in the project management field and they play this course on a time-shared manner along with other activities. The goal of this experience is to increase the competence levels acquired by using an structured web based portfolio tool helping to reinforce how relevant different project management approaches can result for final products and how important it becomes to maintain the integration along the project. Monitoring is carried out by means of have a look on how the work is being done and measuring different technical parameters per participant. The use of this information could make possible to bring additional information to the students involved in terms of their individual competencies and the identification of new opportunities of personal improvement. These capabilities are strongly requested by companies in their daily work as well as they can be very convenient too for students when they try to organize their PhD work.*

Keywords: Project based learning (PjBL); interdisciplinary learning; computer based approach; competence development in project management; industry oriented learning methodology.

1 Introduction

Teaching project management (PM) to graduate engineering students is, most of the times, a challenging matter [1]. This is mainly due to the well-established approach to problem-solving that the student already has developed after years of training on detailed technical problems –very well defined and with only one right solution available–. Leading with this theoretical approach to problem-solving by asking the students to meet client's requirements develops a new approach to problem solving due to the highly undefined nature of the client's requirements. An added difficulty is the length of the course, just 129 hours of student's work (4,8 ECTS), which becomes short time considering the lack of experience of the students[2].

This is not a new problem at all, as different formal approaches have been proposed to cope with it. Problem Based Learning (PBL) has proved to be an excellent method for developing new forms of competencies [3][4]. Research has shown that students retain minimal information in the traditional didactic teaching environment and frequently experience difficulty in transferring the acquired knowledge to new experiences [5].

The actual implementation in several European countries of the new educational model –established by the Bologna agreement– has brought to life a prolific framework of innovative educational initiatives [6].

A Project-Based Learning (called here PjBL to distinguish it from the acronym for problem-based learning) environment enables students to draw upon their prior knowledge and skills, brings a real-world context to the classroom, and reinforces the knowledge acquired by both independent and cooperative group work [6]. A search in the literature shows that the researchers have even found interesting the analyses for estimating the effort of both students and instructors in a competitive collaborative environment based into the PjBL strategy [1][6]. Moreover, specific software tools have been proposed for formalizing the cooperation between teams not located at the same place [7].

It must be ensured that the situation proposed allows multiple solutions, the need of multicriteria decision making processes, enough milestones to consider, and that it involves different technologies and disciplines, etc. In brief, that it complies with the criteria of the CIFTER model [8] to evaluate the complexity of a project.

The main goal of this paper is to present a course configuration that allow the cooperation between different students from different universities, working together in common initiative, allowing the competition between project teams. Indeed, the interest is to go further being able to monitor and measure the management attitudes and capabilities shown by the team members and project management team. This is a key factor in their learning process and makes it a real difference, as feed-back for improvement is given.

The organization of the paper is as follows: in section II the project management competence framework used is presented and globally described. In section III the course itself is presented. According to course characteristics, the software tool implemented and customized is described in section IV. In section V the analysis of the experience is carried out, including the measuring system implemented and the perception of the students with the whole system. Finally, a conclusion section tries to summarize main aspects involved and potential exploitation for additional uses than those presented here.

2 PM competence model

The terms competency and competence are becoming increasingly used by project managers in conversations around selection or development of project managers. Although the twin ideas of competency and competence frameworks first emerged around 25 years ago, their adoption within the project management profession for various purposes continues.

Input – Competencies	Output - Competences	Outcome - Successful project delivery
Typically relate to a wide range of situations	Typically based on a work task and relate to a particular role or job	Performance
Knowledge Skills e.g. decision making ability Personality	e.g. execution of risk management e.g. maintaining stakeholder commitment	Delivered project objectives On course to achieve benefits Stakeholders satisfied
Assessed using PMA5	Assessed using PMA5	Assessed using Scored CV

Figure 1. Relationship between competency, competence and project outcome (source [9]).

So competencies could be considered as the underpinning knowledge, attitudes, skills and behavior that an individual needs to acquire to deliver superior performance. In the case of this paper the interest is to be able to measure competences, as they can be collected from evidences recorded with the project management information system (PMIS).

According to previous concepts, the framework used as a reference for competences was the IPMA Competence Baseline [10]. The IPMA Competence Baseline is the common framework document that all IPMA Member Associations and Certification Bodies abide by to ensure that consistent and harmonized standards are applied. To meet the needs of those interested in the practical application of the ICB, the certification process is described for each level, together with a taxonomy and a self-assessment sheet. Professional project management is broken down into 46 competence elements that cover the following:

- **technical competences** for project management (20 elements);
- **behavioral competences** of project personnel (15 elements); and
- **contextual competences** of projects, programmes and portfolios (11 elements).

Even when length of the course itself is very limited, which imposes restrictions on the dedication of both instructors and students, it was considered a quite challenging experience from different points of view. Only a few of competences are being assessed, but, in any case, the model for competences of project managers is being considered as adequate.

3 Course Description

During the design of the course organization, it was necessary to to consider some facts. First of all, it is the general lack of student's practical experience in the field. Normally, students come with strong technical knowledge for specific technical aspects as they were trained for solving well defined problems or parts of small projects by splitting the scope of the work on an initially well established and not modified plan. Usually, their experience was gained across different courses, by teams of four of five students.

Additionally, it was decided to establish a more realistic framework in different ways such as customer's scope changes or not fully restrained description of the customer requirements. Indeed, the first job of the students is to identify and discuss the scope of the project as well as its deliverables. Also the communication issues need to be properly addressed, not only inside the team but also with other agents of the project. Thus, for example, project teams are requested to deliver a flash video file presenting their proposed solution to the customer's request.

In order to be closer to the reality it was decided to populate the project teams with members (i.e. students) running project management courses at the same time in different universities. This was a challenge because not all the universities start the academic year at the same time, students background is very different (environmental engineers, industrial engineers, mechanical engineers, electrical engineers, etc.), they are located far away in different places (in this particular implementation four courses from three different universities were merged) and they have different internal schedules which raised different organizational difficulties.

Obviously every course at each University has its own programme, according to the specific objectives of the degree involved. That means that different students from different courses and Universities will learn theoretical concepts at different time or speed and the practical work carried out under the presented approach makes possible the reflective learning [11][12] as some of them becomes aware about the reason for some mistakes they performed when the theory is discussed in the classroom.

The shared component for all of these courses is the practical project to be developed. It means about 1400h of work, to be developed during four months by teams of about 40 students each one. The resource allocation always becomes a problem and, even when it is expected that all the students make managerial activities as well as technical work inside those activities, the figure of project manager is selected by self appointment, after a number of surveys per team member were performed. These surveys inform the student about his/her leadership orientation, his/her profile as manager, etc., in such a way that a formal presentation is made, like candidatures to be appointed for project manager by all the teams.

The complete practical work was monitored, troughout regular surveys, as shown for example in Figure 2. Several questions are raised to the students on a two-week basis. Most of them are focused to identify perception about the project, their own role in the team, the workload, and conflict identification or information about properly project development.

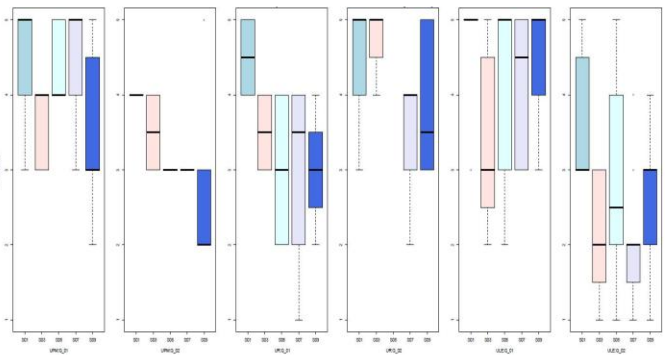


Figure 2. Result of the survey question about the perception of the project climate along the time. Each block is related to one project team. Inside each block two week period perception is depicted.

This way it is possible to identify the underlying model –between multidisciplinary or truly interdisciplinary– that each team implemented [13]. Furthermore, in some cases, it becomes an indicator about internal or organizative problems in the team.

4 The selected software tool

In order to address all the features identified in previous section, regarding the course organization, strong IT skills are required from all the team members, as they are kindly requested to not only perform the technical work but also to update the managerial status of the work, as well as to discuss technical issues of the work being done with other involved team members located in other universities. It is also required that as much activity as possible become available for analyses, in order to provide feed-back to the student about how to improve or how to do better.

Different technical solutions are available but additional factors like cost must be considered carefully, because about 600 students from different Universities per year are involved in this experience and there is no an easy task to have common agreements for buying software licenses available at different places with different IP addresses. Regarding the type of software, as it is required to manage between 10 and 15 simultaneous projects, it seems quite clear that a portfolio management tools is required.

The selected software environment was Project.net (http://www.Project.net) as it has a community version available under open-source license type. The installation was performed in the data center of one of the participants and all the students of all Universities become enrolled on this system.

This software facilitates the students the use of the different roles that coexist in the management of a project, enabling the team members to communicate and work together even though they might be located at distant locations [14].

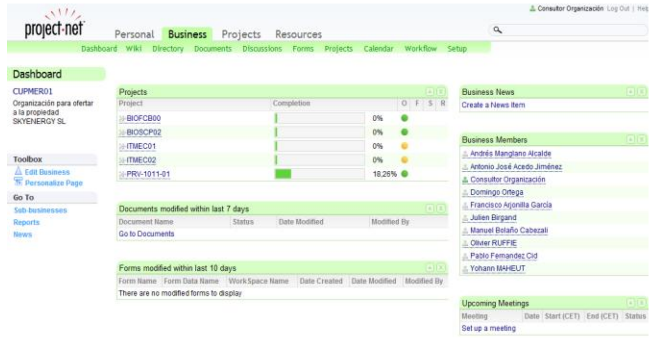


Figure 3. Different projects launched by one team inside one business.

The tool allows presenting a global view about the project, providing to the Project Manager (PM) the main issues requiring actions.

Figure 4. Global view for one running project.

Each student should report, using the software-based support system, the time dedicated to each task, giving as a result the total number of hours the student dedicated to this experience.

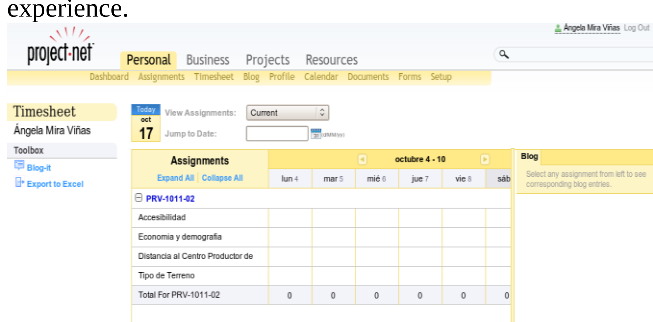


Figure 5. Detail user's view for timesheet control page.

It is possible to make the project plan picture with tasks, relationship and percentage of improvement.

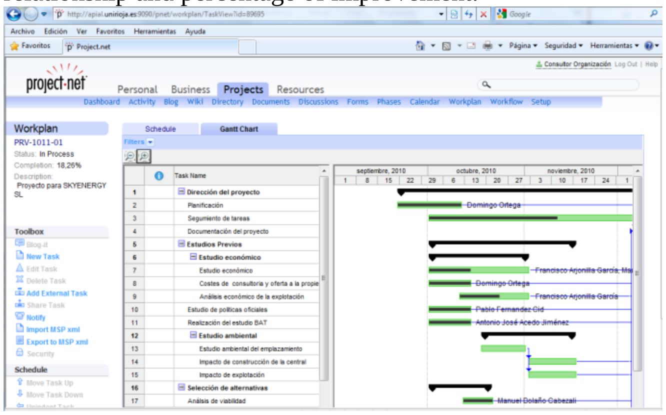


Figure 6. Planning and Monitoring view for the project.

Or even take a detailed look over the project on a task per task approach.

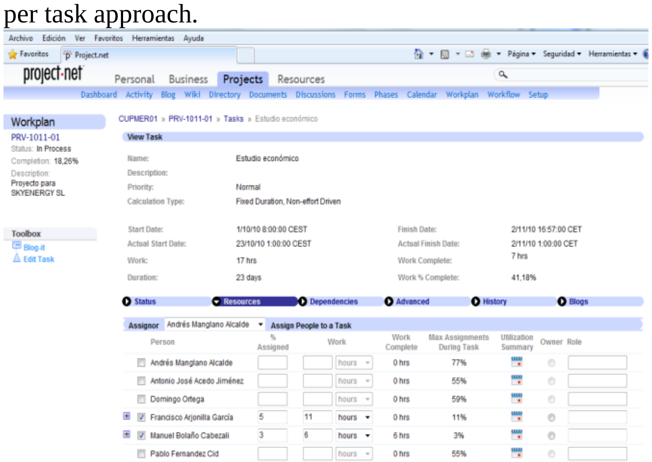


Figure 7. Detail view for one task.

Indeed, for learning purposes, it is possible to carry out a forensic analysis of decisions adopted, as shown in Figure 8. It is possible to review different versions of one document as well as the claims for effort submitted into the relevant task and how the task's product was linked to it.

In addition it is possible to qualify the discussion held between the user of a task's product and their customers (other team members in other relative tasks), as well as the decision made by the project manager.

All that information is regularly collected by the system and it is processed in order to better analyze different student approaches to project management.

Monitoring teachers are in charge of perform these analyses as well as for doing formal audits (twice per semester) where the result, both global for the project as well as detailed per team member is disclosed.

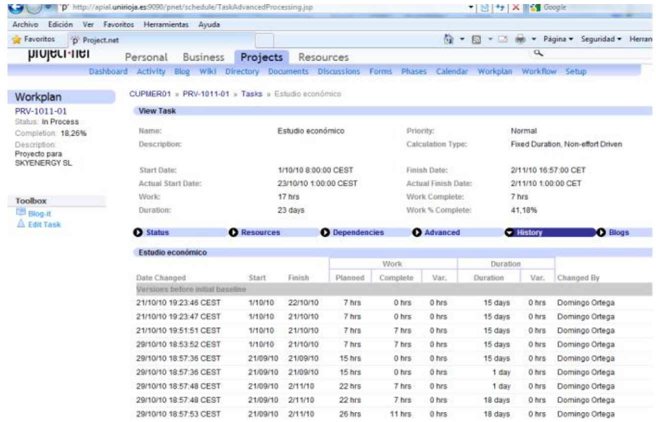


Figure 8. Historic view of changes introduced into a single task. It is useful for forensic analyses.

5 Analysis of the experience

The students acquire the competences not only through the traditional channels but also by the interaction amongst them while using the collaborative tools that Project.net provides. Indeed, there are aspects of the organizational culture that endows the students with a formal work methodology that makes them accustomed to think about what must be done and what effort must be made in order to achieve a specific goal. Moreover, as the deliverables obtained by some members might be inputs in the processes assigned to others, the dependency and connectivity of the task is usually very significant. The software-based support system itself promotes the traceability by allowing multiple versions and complete data.

According to the IPMA competences for project management, the technical competence 1.10-Scope and Deliverables is presented by a Work Breakdown structure (WBS) and its dictionary uploaded as documents to the file directory area. PM is responsible for defining project phases as work packages. For each WP a list of deliverables is identified and declared (see Figure 9). Per deliverable, a list of tasks is identified, and pert task resources are identified and assigned (IPMA Compentece 1.12). All these steps need to be maintained but the system makes possible to monitor its traceability and to assess these PM competences.

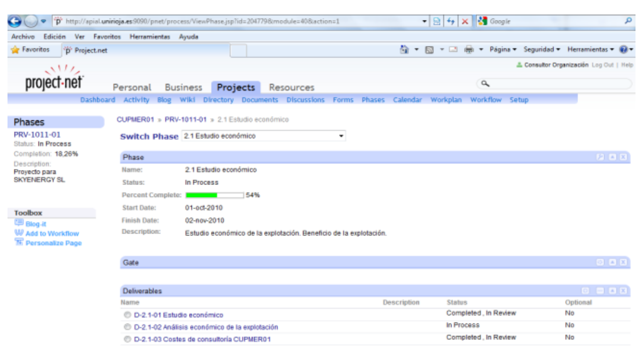


Figure 9. Phase definition with deliverables declared

It is possible to have a detailed view over different parameters along its developing period. Some of them are presented at Table 1, according to the IPMA codification.

Competence		
	Parameters	***Period***
1.02 Interested parties	Number of formal minutes of meetings agreed with customer	4 times
1.04 Project requiremens & objectives	WBS definition, rastreability matrix, List of deliverables and consistent task definition	1 time
1.06 Project organization	Number of WP, Tasks and deliverables	2 times
	Number of work hours initially scheduled and finally claimed	2 times
1.05 Quality	% of deliverables linked to documents and % od deliverables formally approved	6 times
	Averaged quality for deliverables	3 times
1.07 Teamwork	Evolution of standard deviation and average of climea survey.	6 times
1.09 Project structures	Team architecture defined, including formal management and audits	1 time
1.11 Time & project phases	Gannt diagrams and their management, including corrective actions	6 times
1.13 Cost & finance	Negociated budget against number of hours used by team members. Comparison according to EVMS methodology is performed	3 times
1.15 Changes	Amount of PM corrective actions implemented along the project	1 times
1.16 Control & reports	Differnece between reported situation and evidences coming form the software tool by auditing processes	3 times
1.17 Information and Documents	Perception between the selfassessment and the evaluation carried out by product users	3 times
1,18 Communi-cation	Quality of messages delivered by the video as well as its organziation and resources used	1 time

Table 1. List of Parameters related to technical competences monitored according to this proposal

Indeed many other competences are considered but less formal monitoring approach is used, like self-control, motivation, negotiation creativity and ethics.

On these competences only indirect evidences are recorded and, because of that, only suggestions are raised as feed-back to the participants.

Different phases can normally be observed during the experience. Initially, students do not believe possible to manage such under-defined proposal and they look for common understanding of the problem because they come from different environments. During this phase several meetings are held with the customer (some teachers playing that role).

Next, they face learning problems regarding detailed management of the platform and specific coaching actions from the project manager team is required to put everyone on the rails.

Later on, student teams realize that some of the initial hypotheses made are not longer valid and scope redefinition is requested (change management) but the deadline is fixed. Because of this, the pressure increases and the stress too. These effects are amplified because of the distance and the different background of the team members. This period is quite convenient in order to monitor the leadership qualities of the project manager.

Sometimes, because the differences in course organization in different universities it is possible to identify wrong approaches to different issues. In these cases, scenarios of learning from mistakes occur and it becomes necessary to carefully explain to them that this effect is, in fact, an interesting opportunity, because they become really afraid about wasted time.

Regarding the experience itself, students scored it really well as shown in Figure 10. In this figure all the student answers from different degrees and Universities were considered and answers were graded from 0: fully disagreement to 5: fully agreement. The anonymous survey is runned once per semester and the last week of the course. This survey also is focused on perception about their competence adquisition during the course.

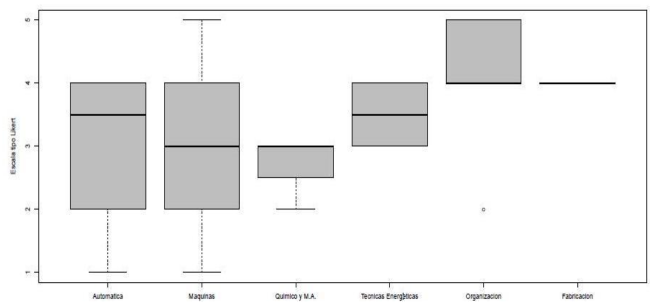


Figure 10. Student perception for the presented experience. Groups are for different student's degree and asked question was "Do you feel the practical experience of the PM course provides you with valuable skills as well as with toolset and knowledge useful in the industry?"

A number of additional questions were surveyed helping teachers to get better understanding about the student perceptions as well as about how to improve the experience. In this particular edition (course 2010-2011) a clear demand was to increase the managerial activities for all the participants, instead of concentrate them in the project management team.

A learn lesson from the teachers team was try to produce a common view for the evolution of the different projects and, in order to making it happening, additional and automatic reports are requested in order to have more common an detailed views for the project. This will produce a more detailed and evidence supported report status which will improve the unification of diagnostics, instead on being dominated for distinct and different perceived particular aspects.

6 Conclusions

The experience presented here show us how, by using computers as a common tool, different cooperation initiatives between different students in different universities and in different places work all together under a collaborative learning approach. This collaborative effort is just a new way of learning by doing and it is combined in different ways at every participant university.

In spite of the lack of previous experience of participants and the limited time frame available very good organizing capabilities have been acquired by participants, regarding not only the rubrics for every curse but also the parameters established for monitoring the performance on specific project management competences. These competences are the key ones for practical work at the industry according the IPMA association and that makes an added value to the initiative.

The individual effort is monitored as well as the decision made inside the project and explanation according the theory is sometimes requested. Different challenging but common factors are also addressed, like multicultural resources, 'virtual' teams, real scope including scope changes and time & budget constraints. Feedback is provided to the participants regarding the auditing capabilities of the software tool selected and the parameters identified as relevant. Additional surveys regarding project climate and feelings are carried out in addition to the normal work.

As main conclusion, students support the experience as they perceive like a reality immersive approach allowing them to acquire specific skills. In addition they remark the adoption of specific work methodology allowing them to integrate the work from others. To this particular field they agree with the tool used as they find it has a friendly web interface for collaborative work as well as powerful enough for helping them in the project management.

Teachers found very convenient the auditing information provided by the software as they are able to provide feedback to the students regarding that specific data. In this area it was found very convenient to measure individual parameters as indicators for student's competence performance. Specific statistical analyses will be possible in the future, when enough data become available.

It is found that this methodology becomes useful for students not only at graduate level, and not only for project teams but also for specific research projects. Additional analyses regarding added value is being performed before extend its use to PhD students, in an effort for providing them with a specific methodology for management along their own path but also in order to collect systematic information for the process itself. In this case a combined approach between process improvement and data mining could be beneficial because it can provide detailed as well as statistically relevant information about problems or bottlenecks or even wrong designed subprocesses.

Acknowledgment

Authors like to thank the funding of the initiative from Spanish Education Ministry throughout the program "Studies and Analyses". The grant reference is AE2010-0001.

7 References

[1] Cobo-Benita J.R., Ordieres-Meré J, 2010. Learning by doing in Project Management: Acquiring skills through an interdisciplinary model. IEEE EDUCON Conference.

[2] Warfvinge P. (2008). A generic method for distribution and transfer of ECTS and other norm-referenced grades within student cohorts. European Journal of Engineering Education. v 33(4). pp:453-462.

[3] Graaff E., and Kolmos A., (2003). Characteristics of Problem Based Learning. International Journal of Engineering Education. 19(5), pp. 657-662.

[4] Kolmos A. and Kofoed L. (2002). Developing process compelencies in cooperation, learning and project management. Proc. 4th World Conference of ICED.

[5] Ivanitskaya, L, Clark, D., Montgomery, G. and Primeau, R., 2002. Interdisciplinary Learning: Process and Outcomes, Innovative Higher Education, 5(2), pp. 95-111.

[6] Schmidt H.G. 1993. Foundations of problem-based learning: Some explanatory notes. Medical Education, 27, pp. 422-432.

[7] Crespo E.,González-Marcos A.; Alba-Elias F; Castejón-Limas M.; Ordieres-Meré J, (2011). Project Management learning in a collaborative distant learning context. An actual ongoing experience. CSEDU-2011. Neederlands.

[8] Global Alliance for Project Standards. 2006. A Framework for Performance Based Competency Standards for Global Level 1 and 2 Project Managers.

[9] Delo A., Hepworth P, Hepworth A. (2010) Assessing the competent project manager. http://www.provek.ac.uk.

[10] Caupin g., Knoepfel H., Koch G., Pannenbäcker K., Perez-Polo F., Seabury C. (2006). IPMA Competence Baseline, version 3. International Project Management Association.

[11] Boyd E.M., Fales A.W. (1983). Reflective Learning: Key to Learning from Experience. Journal of Humanistic Psychology, Vol. 23, No. 2, pp. 99-117.

[12] Turner, J. R., & Müller, R. (2005). The project manager's leadership style as a success factor on projects: A literature review. Project Management Journal, 36(2), 49–61.

[13] Borrego M., and Newswander L.K. (2008). Characteristics of successful cross-disciplinary engineering education collaborations, Journal of Engineering Education, 97(2), pp. 123–134.

[14] Moursund D., 1999. Project-based learning using information technology. Eugene OR, International Society for Technology in Education, pp. 132-143.

Traditional and new materials: A good cocktail for the Mathematics' learning in Engineering Schools.

Ana Belen Cabello[1], Angel Martin del Rey[2], Gerardo Rodriguez[3] and Agustin de la Villa[4]
[1], I.E.S. Joaquin Araujo, Madrid, Spain
[2]Applied Mathematics Department , E.P.S Avila, Salamanca University, Avila, Spain
[3] Applied Mathematics Department , E.P.S Zamora, Salamanca University, Zamora, Spain
[4] Applied Math. Department, Pontificia Comillas University and Polytechnic University, Madrid, Spain

Abstract - *We begin by offering some general considerations about University studies in Spain, guided by the Bologna process. Owing to the necessary changes, we make a proposal about the materials to be used for Mathematical courses in Engineering Schools and the main characteristics that each must meet. On the basis of such considerations, we analyze a teaching strategy, which should be a cocktail containing good "drinks" (materials) with an excellent combination of such "drinks" and excellent "barmen" (teachers and instructors). We conclude with an example of a subject -Linear Algebra- in which the considerations described above have been carried out.*

Keywords: CAS, b-learning, e-learning, Linear Algebra.

1 Introduction. The Bologna process in Spanish Universities.

The Bologna process, which has just been implanted in Spanish Universities in 2010-2011, has involved a profound change in University teaching practices. For more information see European Commission, Directorate-General for Education and Cultures (2009) "ECTS users' guide". Accessed via http://ec.europa.eu/dgs/education_culture/publ/

The milestones of the process are:

i) Teaching based on competences instead of an accumulation of concepts, which was typical of the earlier curriculum. (See Brown, R.B et al [3], Miguel, M. [12], Mulder, M. et al [13], Zabala et al [17])

To achieve this, it is necessary to establish the syllabus of the subject, providing students at the beginning of the course with information about a series of **generic** competences (common features for all mathematical topics in their curriculum), such as self learning, capacity for analysis and synthesis, planning and organization, communication and writing, communication and mathematical expression, science and technology, use of communications technologies, critical thinking, etc., and **specific** competences, such as knowledge, understanding and using the mathematical concepts in Linear Algebra, Calculus in one and several variables, Differential Equations, etc., the ability to apply knowledge, computation and technology to solve mathematical problems for Engineering, etc.

These competences should appear in **learning results**, which are those that should set the guideline for designing the subject.

ii) Organization of students' work, attempting to optimize with this the time assigned to the study plan (1 credit means approximately 30 hours of students' work).

This aspect is of utmost important for the success of the process. Until now, the only work demanded of students was to pass a **traditional** exam at the end of the academic year successfully, although it has been increasingly more frequent to offer periodic exams (along the semester) to assess the progress of their teaching and the students' learning. The exams comprised a series of problems and issues to be solved by the students in a short period of time.

iii) Monitoring of student's learning processes, through activities carried out along the period in which the mathematical topic is taught. For this, it is necessary to have a careful planning of such activities, attempting to make them of different types (traditional exams, laboratory sessions, group work, modeling, and the resolution of engineering problems, etc) so that they can cover and evaluate the competences and learning results.

In sum, the Bologna process demands a teaching style with clear aims and learning results, monitored by a series of varied activities along the whole of the academic period. One example of Calculus course according Bologna process can be shown in Garcia et al. [11].

2 The materials for designing the course.

Careful planning of the course: concepts to be addressed, estimated time allotted to each, the way of delivering them, etc, all lead to a choice of materials that will be useful for students. The choice of such materials is very important for ensuring success in their learning.

Essentially, the materials to be used are the following:

A **textbook**, whose choice should aim at being especially adequate for the students. Some examples of appropriate text books could be Diaz, A et al [5], Garcia, A. et al. [7].

Files for the application of CAS. These files should reinforce, and in some cases could substitute, the traditional teaching of theoretical concepts. The CAS used can be chosen according the needs and possibilities. The most popular are: DERIVE, MAXIMA, MATHEMATICA, MAPLE, MATLAB, etc. Examples of these files can be shown in Garcia, A. et al. [7].

A learning management system (MOODLE or equivalent), which in the case of face-to-face University teaching should aim at allowing students to access the materials easily and which can also be used as communications tools (chats, forums, E-mail) and for the management of self-assessment tests or as a revision tool, conforming a battery of tests or problems that are offered to the students on a “random” way.

Some examples of use can be shown in Polytechnic School of Zamora in the document “Teaching guides”. http://campus.usal.es/~guias2009/centros/guiagrado.php?id=21, Rodriguez, G. [14] and Rodriguez, G. [15].

The choice of some **web sites**, selected by the instructor that could contain: examples of the application of the mathematical contents, visualization demos, mathematical computations, the possibility for experimenting with problems offered randomly to students, etc.

3 The teaching strategy.

It is undoubted that there has been a prominent evolution in the teaching of mathematics. Twenty years ago, the teaching of this subject was reduced to an ordered collection of definitions, examples, theorems, corollaries and problems. Currently, however, it could be said that we are almost at the opposite extreme. At the present, Engineering Schools –in general- are diminishing theoretical concepts, and the teaching is focused in problem-solving without the necessary background to understand the problems.

It is not possible to understand the teaching of maths without theoretical concepts, which students must understand and be able to handle. This is because the role of mathematics in Engineering Schools should be formative (mathematics allow one to “furnish” the heads of the students) and also informative, providing the mathematical concepts that will later be used in more technical subjects of their curriculum.

We must also be able to bring mathematics closer to our Engineering students, showing them -as far as they are able to understand- cases of Engineering that they will be able to handle with the mathematical knowledge they have acquired. The teaching strategy should be aimed at optimizing students’ work, fully bearing in mind that their work load should be reasonable within their circumstances since it would be stipulated by the credits assigned to each mathematical theme. In the current University structure, most subjects are given on a semester basis with 6 credits (180 hours of work/student, of which 60 must be followed in face-to-face classes). Accordingly, the obligations of the instructor are to “generate” a work that will take students approximately 2 hours to do per face-to-face hour of class received.

In light of the above, we propose the following:

Textbooks. These books could be considered as **musical scores** that the instructor will interpret according to the needs. Their structure is that of an exposition of theoretical contents followed by a self-evaluation test, solved problems and unsolved problems, sequenced in the order of contents and difficulty. These books will act as guides for theoretical and practical (problem-solving) classes.

A balanced use of CAS. To accomplish this, the first task to be undertaken is to choose an appropriate CAS, which to a large extent will depend on economic and logistical possibilities, inherent to the organization of the course. Theoretically, it would be possible to consider two types of CAS: Simple ones, such as DERIVE and MAXIMA, and more powerful ones, such as MATHEMATICA, MAPLE and MATLAB. In the case of simple CAS, we propose that, after having elaborated simple tutorials regarding their handling, which that must be studied by the students previously, they should be included in traditional lectures in visualization tasks, heavy computations, experimentation, etc. In the case of the more powerful software, we must design sessions in a Mathematics laboratory. Different possibilities about the use of CAS can be shown in Garcia et al [8], [9] and [10].

Reinforcement tasks (mainly the most relevant issues appearing in the text book).

Solution of real problems, in keeping with the **problem-based learning** strategy.

Students must model the problem, apply the required mathematical concepts, and solve the problem with exact or approximate techniques. All this must be done under the supervision an instructor who will set and tutor the different steps in the process of solving the problem. See Cazes, C.; et al. [4].

All the information will be transmitted through a learning management system, in which students will be able to access chats, forums and other communications activities and assessment tasks.

4 An example.

We expose the main characteristics of a course adapted to the parameters quoted in the previous paragraphs. This example is devoted to the Linear Algebra topics.

A standard Algebra course in an Engineering School should have contents similar to those specified below:

i) Algebra´s basic tools: Matrices. Systems of linear equations. Determinants.

ii) Vector spaces: linear dependence and independence. Vector subspaces. Basis of a vector space. Dimension. Coordinates. Change of basis.

iii) Linear transformations: Definition and properties of linear transformations. Matrices and linear transformations. Eigenvalues and eigenvectors. Jordan’s canonical form.

iv) Euclidean spaces: Inner products. Orthogonality. Projections. Least squares method. Orthogonal diagonalization. Orthogonal transformations.

The materials to be provided to the students are as follows:

Textbook: The course is based on the book Villa, A [16], recently published. The book contains: theoretical results, theoretical questions, solved and proposed problems. This book also contains a CD with files (produced with different software packages: DERIVE, MAXIMA and MATHEMATICA) showing the different possibilities for using CAS in a Linear Algebra course.

CAS: We integrate the CAS in the teaching, since they allow students to experiment with different situations because they do not have to make tedious calculations by hand and they can solve problems that are closer to real-life situations and not only canonical problems with the results prepared.

We promote the use of the following:

-By the instructor in class with different aims: demos involving graphical results, computations, problem-solving, etc.

-Laboratory sessions in which the students can do **computer exercises** about the knowledge already gained in theoretical lectures.

-Tutorials that the students have free access to and that contain explanations of the use of the CAS in a general way or the CAS commands that they will have to use.

Depending on the type of software, its use will be different since software with more features, such as MATHEMATICA demands a more rigid syntax and hence it will be necessary to employ more time to gain the use of the CAS in a fluid way.

Accordingly, the introduction of the CAS is performed through **tutorials** through the following files:

- A tour, which analyses the possibilities of using MATHEMATICA in a general way.

- General concepts that provide insight into the MATHEMATICA commands useful for working with vector spaces, matrices, linear applications, determinants and sets of equations.

- Autovalvect, which explains the commands of MATHEMATICA that are useful for studying the theory of eigenvalues and eigenvectors.

- Euclidean Spaces, which shows the main MATHEMATICA commands related to Euclidean space and orthogonal transformations.

The use of MATHEMATICA would imply that some sessions should be devoted to introducing it since after many years of experience using it in the field its syntax and its use in general advise this. Once students believe that they are able to handle the MATHEMATICA commands, they are recommended to solve the problems usually addressed in class by hand with the help of the CAS. As an example, all the problems proposed in the text book have been solved using MATHEMATICA.

If MAXIMA is used, which at the present time of budget restrictions is advisable since it is **free software**, see Abanades et al. [1] there is no need to devote any special sessions to explain its use because the syntax is very easy. The students merely receive tutorials: matrices, sets of equations, eigenvalues and eigenvectors and Euclidian space, explaining in the tutorials the commands of MAXIMA that allow problems related to the concepts analyzed in the tutorials to be solved. The instructor uses MAXIMA integrated in traditional classes and students must do certain exercises of those proposed in the work tasks using MAXIMA.

In the CD accompanying the book there are several files including the solution (using Maxima) of many of the problems proposed in the different chapters of the book.

Electronic material: Certain web-sites (not many) showing applications, the capacity to experiment, etc., are provided to the students. One of them, for example, would be: http://aix1.uottawa.ca/~jkhoury/app.htm

Projects: One part of the assessment is the development of a Project in groups of 3-4 students. A list, not very exhaustive, offered to the students in the 2010-2011 academic year is as follows: Kirchoff´s laws for electric circuits, discrete dynamic systems, matrix factorization methods, matrices and cryptography, magic squares, the distribution of temperature on a plate, the Fibonacci sequence and the golden number, Leontief models, applications of the spectral theory: Genetics, population growth.

In each model, the aims and bibliography (traditional and through the internet) are provided to the students and they must report their results to their peers.

With the materials, a deep organization of the work must be prepared. The students, at any time, must know what they should be doing, how to use the CAS, how to approach the project assigned, etc. And all this bearing in mind that the students' work-load must be as stipulated. And obviously also the assessment method according the course's design must be explained carefully. See Bokhove, C et al [2]. Let us show an example of this kind of projects (see [6]).

A surveillance device has access to the images from a security CCTV that focuses on four sides of a building. (See figure 1). The device is programmed in such a way that it only shows one of the sides of the building on the screen. After showing the same side for one minute, it may "choose" to maintain the image from the same camera, with probability a ($0 \le a \le 1$) or may access one of the two contiguous sides of the building, with an equal probability, which would be (1-a)/2. The security guard controlling the device introduces the value of a. See figure 1.

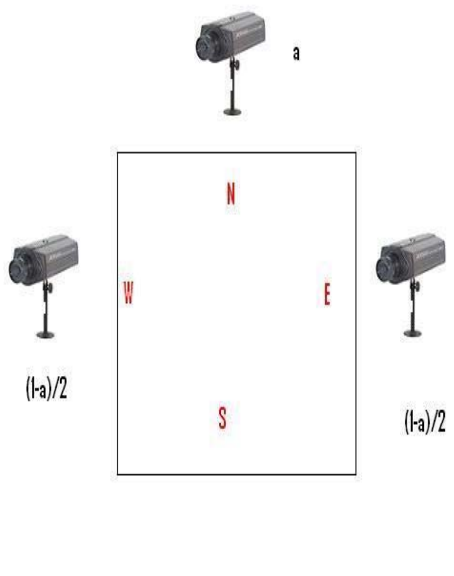


Figure 1: Surveillance device

i) Which value of parameter a should be introduced for displaying the same side of the building constantly?

ii) At 8:00 a.m. the device displays the Nord side. The guard introduces the value a = 1/2. Find the probability of showing each of the sides at 9:00 a.m. Analyze the same problem with different values of parameter a. Pay special attention to the cases a =0 and a =1.

iii) Study, for different values of a, the behavior of the device, after n minutes (with n very high).

iii) Study the same problem for different sides of the building n, distinguishing between n odd and even in the case a =1.

5 Conclusions

Bologna process has meant a profound change in University teaching, to which the teaching staff is now becoming adapted.

The characteristics of this process mean that the global work-load of the students must be suitably programmed.

To program the students' work-load the course materials must be selected with great care. We propose a mixture of traditional materials (textbooks and collections of problems) – with others that could be called e-materials, such as CAS, WEBs, etc., managed through a Learning Management System.

We also believe that students are moderately satisfied with the process, although these are early days and it would be imprudent to attempt to draw definitive conclusions, since we are still in the process of adaptation.

6 References

[1] Abanades, M. A.; Botana, F. Escribano, J. and Tabera, L. F. *Software matemático libre*. La Gaceta de la R.S.M.E. V.12, n.2 p. 325-346. 2009.

[2] Bokhove, C., Drijvers, P. Digital tools for Algebra Education: Criteria and Evaluation, *International Journal of Computers for Mathematical Learning*, **15**(1), 45-62. 2010.

[3] Brown, R.B.; McCartney, S. *Competence is not enough: meta-competence and accounting education*. Accounting Education, v. 4, 1, p. 43-53. 1995.

[4] Cazes, C.; Gueudet, G.; Hersant, M. and Vandebrouck F. *Using e-exercise bases in Mathematics: Case*

studies at University. Int. Journal of Computers for Math. Learning , 11, p.327-350. 2006.

[5] Diaz, A.; Hernandez, E.; Tejero, L. *Algebra para ingenieros* SANZ Y TORRES. 2010.

[6] *Diaz, A.; Garcia, A,.; Villa, A. de la.* An example of competence-based learning: Use of Maxima in Linear Algebra for Engineers. To appear in the International Journal for Technology in Matematics Education.

[7] Garcia, A. et al. *CALCULO I. Teoría y problemas de Análisis Matemático en una variable.* Third edition. CLAGSA. 2007

[8] Garcia, A. (Ed) *Prácticas de Matemáticas con Derive*, Madrid: Clagsa 1994 (or *Mathematisches Praktikum mit DERIVE*, Addison-Wesley, (Deutschland), 1995

[9] Garcia, A., Garcia, F., Rodriguez, G., De la Villa, A. Some unexpected results using Computer Algebra Systems. *The International Journal of Computer Algebra in Mathematics Education,* **8**(3), 239-252. 2001.

[10] Garcia, A., Garcia, F., Rodriguez, G., De la Villa, A. A toolbox with DERIVE, *Proceedings of ACA 2009, 15th International Conference on Application of Computer Algebra, Montreal, Canada.* 2009

[11] Garcia, A., Garcia, F., Rodriguez, G., De la Villa, A. Calculus in one variable: One Spanish overview according EHEA. *15th SEFI MWG Seminar and 8th Workshop GFC Mathematical Education of Engineers, Wismar, Germany*. 2010.

[12] Miguel, M. *Modalidades de enseñanza centradas en el desarrollo de competencias: Orientaciones para promover el cambio metodológico en el Espacio Europeo de Educación Superior*. Universidad de Oviedo. 2006.

[13] Mulder, M., Weigel, T. and Collins, K. *The concept of competence in the development of vocational education and training in selected EU member states. A critical analysis,* Journal of Vocational Educational and Training, 59, 1, p. 65-85. 2006.

[14] Rodriguez, G. “Un proyecto europeo para la enseñanza de las matemáticas: el proyecto EVLM”. In *II Jornadas de Innovación Educativa de la EPS de Zamora*. Zamora. 2007.

[15] Rodriguez, G. “El Centro de Matemáticas de la Universidad de Salamanca”. In *I Congreso Internacional de Intercambio de Experiencias de Innovación Docente Universitaria*. Salamanca. 2008.

[16] Villa, A. *Problemas de Álgebra*. CLAGSA. 2010

[17] Zabala, A., Arnau, L. *Cómo aprender y enseñar competencias*. Ed. Grao. 2007.

Genome Model – A context on Measurement of Learning Effectiveness by Aligning Objectives and Assessment

Dr. Subraya B. M.[1], Manjunatha Prasanna S.[2], Amit Purohit[3]
[1]Associate Vice President and Head, Global Education Center, Infosys Technologies Limited, Mysore, INDIA
[2] Principal, Global Education Center, Infosys Technologies Limited, Mysore, INDIA
[3] Member, Global Education Center, Infosys Technologies Limited, Mysore, INDIA

Abstract: *Assessment is an indispensable part of any Education Center. Assessment is needed to measure the learning effectiveness. Besides measurement, assessment helps to validate the other dimensions of the Education Center like effectiveness of knowledge building, knowledge dissemination and knowledge management. Measurement of learning effectiveness poses challenges, as a chasm exists between the identified course objectives and the assessment. The gap widens even more in context of assessing the people learning Computer Science and Computer Engineering, as the field is evolving at a rapid pace. This drives for arriving at a model to give a holistic view to the design of assessment. The current study focuses on the corporate training and the design of a model to bridge the gap between the objectives and the assessment. The study also attempts to measure the learning effectiveness through assessment.*

Keywords: *Course Objectives, Learning Outcomes, Assessment, Seed Questions, Genomes.*

1. Introduction

The process followed in an Education Center as represented by Figure-1, identifies the various stages through which the assessment is being derived. Each Education Center operates with a central objective which helps to set a context for designing a package for necessary knowledge building. Package is realized in the form of a program which aids to execute the knowledge building process. Program in turns decides the courseware. The Education Center objectives thus get transformed in the form of course objectives.

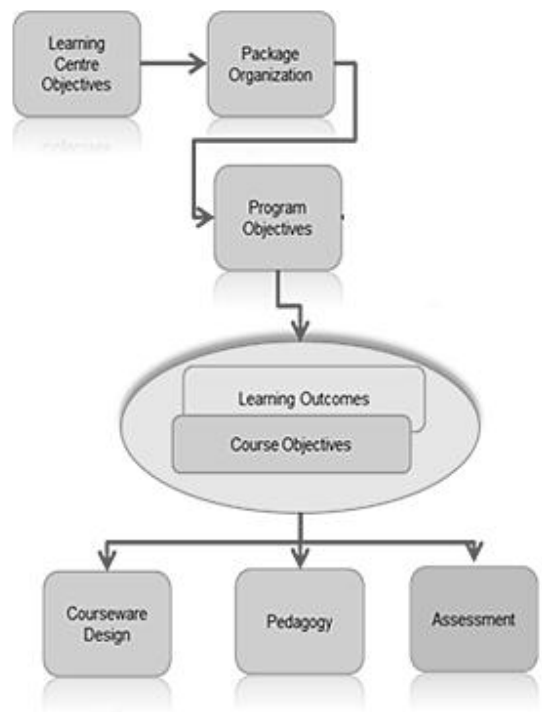


Figure 1: The Education Center Process Map

The courseware design and the pedagogical mechanisms are also important dimensions of an Education Center program; but in this paper we will limit ourselves to the design of assessment, although the other two dimensions affect the assessment.

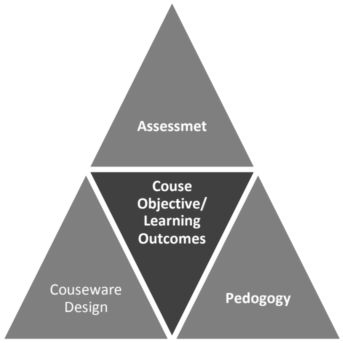


Figure 2: The Dimensions of a Corporate Training Program

In a typical corporate training program, the expected learning outcomes change more rapidly than an academic program, as the objectives for corporate training program are derived from the real time business requirements, organizations policies and the global factors. Under all such circumstances the assessment for a corporate training program is influenced by various factors viz.:

- Tangible factors
  - Course scope
  - Scale and Diversity of the learning community
  - Limited course duration
  - Accepted repeatability of the assessment questions

The above factors are all measurable and hence the corresponding metrics could be used to design the assessment. Moreover the compelling requirement is the continuous improvement to the assessment in view of all above tangible factors. Continuous improvement demands a loopback system to provide a feedback to assessment design based on the achieved learning outcome. Measuring achieved learning outcome from an assessment is one of most challenging aspect posed by competing factors like:

- Multiple learning objectives
- Lack of relative ranking of the objectives
- Poor assessment design etc.

Measurement helps us to identify the learning effectiveness and the feedback in sent back to iterate the process. So, there is a need for a model to bridge expected learning outcomes and assessed learning outcomes.

2. Base Formulation

Typically, the central objective of a corporate training program is identified through a detailed business case analysis, empowered by the vision of the organization. During business case analysis, various competing factors influencing corporate training are determined and evaluated leading to objectives of Corporate's Education Center. The Education Center objectives need to be organized in the form of various learning packages conforming to various learning domains. To realize the training, packages provide necessary scope to arrive at programs as per the training needs.

2.1.Definition of Terms

2.1.1 Education Center Objective

Is the high level descriptor of the organization's vision towards knowledge building and learning. The objectives are categorized as per learning domains and organized in the form of packages. Formally, the Education Center Objective (LCO):

$$LCO = \{PC1, PC2, PC3\dots PCi\} \qquad (1)$$

Where, PC represents the various packages in which the objectives are organized.

2.1.2 Package

A package is a logical grouping of learning domain areas. Various programs can be derived out of one or more packages aligning with the program's objective.

Formally, a Package:

$$PC = \{D_1, D_2, D_3,\dots, D_j\} \qquad (2)$$

Where, D represents the various learning domain areas grouped in a package.

2.1.3 Program

A program is identified as an ordered collection of courses, realizing the learning domain areas derived from one or more packages and helps to delimit the scope of the learning domain. Program objectives are derived from the Education Center objectives. Formally, a program (P):

$$P = (C1, C2, C3,\ldots, Ck) \quad (3)$$

Where, C represents courses realizing the various learning domain areas. A program is characterized by various properties like:

- Need to have a specific duration and scope
- Structured execution
- Measurable output

2.1.4 Course

A course can be identified as collection of logically structured concepts for which a learning outcome can be derived. Course objectives are derived from the program objectives. For measuring the learning effectiveness, we have considered that each course objective is associated with a unique learning outcome.
Formally, a course (C):

$$C = \{CO_1, CO_2, CO_3,\ldots, CO_k\} \quad (4)$$

Where, CO represents identified course objectives in a course. The relationship between the course objectives and the associated learning outcome can be represented as a bijective function [2],

$$f: C \rightarrow L \quad (5)$$

Where, C is set of course objectives and L is set of associated expected learning outcomes.

2.1.5 Assessment

Assessment is a mechanism to measure the extent to which the course objectives are achieved. More formally, assessment measures the extent to which the expected learning outcomes are met.

2.1.6 Genomes

Genomes exhibit the essential characteristics of the course objectives. Genomes carry the syntactic and semantic primitives of the course objectives. In this study we consider syntactic primitive as 'Explicit Genomes' and semantic primitives as 'Derived Genomes'.
Formally,

Explicit Genome : $EG_i = \Phi$ (CO), and
Derived Genome : $DG_i = \psi$ (CO)

Where, Φ is 'Word Stemming Function' [6] which extracts keywords from the CO and the ψ is semantic extraction of the encoded primitives of LO in CO.

2.1.7 Seed Question

Seed question encompass one or more genomes as a template question outlining one or more identified learning outcomes with a notion of generalization.
Formally,

$$S_Q = \gamma(EG, DG) \quad (6)$$

Where, γ represents 'Synthesis' over Explicit Genomes and Derived Genomes.

2.1.8 Minimal Set of Seed Questions

It is a subset of S_Q encompassing all the identified genomes at least once. Formally,

$$S_{QMin} \subseteq S_Q \quad (7)$$

Such that,

$\forall$ i : (EG_i, DG_i) are encompassed by S_{QMin}

Clearly, $S_{QMin} \leq$ Total number of genomes

2.1.9 Rank of Seed Question

It is defined as the total number of genomes encompassed in the seed question. Formally,

$$S_{QRank} = \textit{Number of genomes in } S_Q \quad (8)$$

2.1.10 Derived Question

Derived questions are used in the assessment to assess learner on the learning outcomes. More formally, a derived question is realized by specialization of the seed question in the context of the expected learning outcome, viz.

$$S_Q \rightarrow D_Q \quad (9)$$

Where, D_Q represents Derived Question and $\rightarrow$ represents the specialization.

2.1.11 Core Assessment

Core Assessment includes the minimal set of seed questions or the corresponding derived questions.

3. The Model

3.1. Components of the Model

1. Course Objective(s) : *CO*
2. Learning Outcome(s) : *LO*
3. Genome(s) : *EG*, *DG*
4. Seed Question : S_Q
5. Minimal Set of Seed Questions : S_{QMin}
6. Derived Question : D_Q

3.2. The Process

The process of arriving at assessment from the course objectives is depicted in Figure 3.

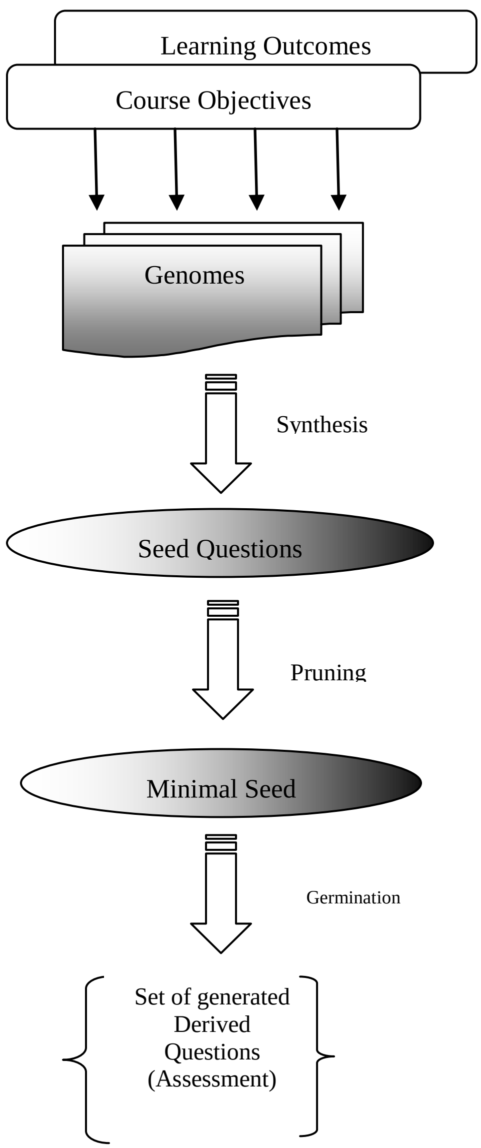


Figure 3: The Process Involved

3.2.1 Identification of the course objective(s)

An objective describes the intended purposes and expected results of the training and establishes the foundation for assessment [3]. The identification of the course objective is not part of this study.

3.2.2 Identification of the expected learning outcome(s)

Learning outcomes are the statements that describe significant and essential learning that learners have achieved and can be reliably demonstrate at the end of a course or a program [3]. In this study we have adopted the guideline for arriving at the course objectives and learning outcomes as suggested by Eric [3].

3.2.3 Identification of the Genomes

In the proposed model the Genomes are identified as “Explicit” and “Derived”, by word stemming [6] and semantic analysis respectively. Various keywords from the objectives are identified and taken apart; these keywords form the Explicit Genomes. The Derived Genomes are created by semantic study of the course objectives, supported by domain knowledge. The course objectives are likely to be simple and precise, and may fail to project implicit domain terminologies. Semantic study is used by the domain experts to associate Explicit Genomes with implicit domain terminologies resulting in Derived Genomes. Moreover, for a course we can classify the genomes in three categories based on the expected level of competency as given in Table 1.

Table 1: Classification of Genomes

Category	***Expected Level of Competency***
Must to Know	Masterly Skillset
Need to Know	Essential Skillset
Good to Know	Exposure Skillset

3.2.4 Synthesis of Seed Questions

The seed questions are synthesized as *a template* in the form of an interrogative statement, encompassing one or more genomes addressing at least one expected learning outcome.

3.2.5 Pruning of Seed Questions List

The seed questions are arranged in a list in the decreasing order of S_{QRank}. Simple top-down pruning over the list until all the genomes are covered results in S_{QMin}. Clearly, S_{QMin} leads to Core Assessment.

3.2.6 Germination of the Derived Questions

The derived questions are germinated as an interrogative statement from seed question template by substituting the generalized aspects within the context of the course scope. Clearly from a single seed question many derived questions can be germinated.

3.3. Property of Completeness

One of the challenges implicit to the design of the assessment is the assertion about the coverage of all the course objectives. In this model, we propose the property of completeness to address this challenge. The assessment is said to be complete if and only if, assessment at minimum is a Core Assessment.

3.4. The Metric

One of the challenging aspects of any learning process is the measurement of the learning effectiveness. In this paper we are deriving the learning effectiveness as a function of various attributes like:

- Number of participants participated in the assessment
- Number of assessment questions
- Number of identified genomes etc.

For arriving at the learning effectiveness, we've defined following three metrics:

3.4.1 Learning Index (Q_{LI})

Learning Index identifies the assessment metric determined at the course design level depending on the expected level of competency of the identified genomes. Clearly, the Learning Index sets the benchmark for the expected level of competency for a course and hence is absolute for a course.

More formally, the *Learning Index* defines the quality of the assessment for a course as a function of the identified genomes.

$$\text{Learning Index} = 1/X \sum_{i=1}^{3} (\text{Wi}\, Xi) \qquad (10)$$

Where,
X_1 = Number of Must to know genomes
X_2 = Number of Need to know genomes
X_3 = Number of Good to know genomes
W are the competency level coefficients for identified genomes, ranges from 0 to 1
and,

$$X = X_1 + X_2 + X_3 \qquad (11)$$

Weight factors for the genome are derived from the course objectives and the learning outcomes. As, the *Learning Index* is the weighted average of the competency levels, its value ranges from 0 to 1.

3.4.2 Assessed Index (Q_{AI})

Assessed Index identifies the quality of the learning by outlining the quality of the assessment result. In this study we've identified the Assessed Index as a factor of:

1. Number of learners (N)
2. Number of seed questions in S_{QMin} (T)
3. Number of questions answered correctly (C)

$$\text{Assessed Index} = 1/N \sum_{i=1}^{N} \left(\frac{Ci}{T} \right) \qquad (12)$$

As, the *Assessed Index* is derived as an average over number of learners (N), its value ranges from 0 to 1.

3.4.3 Learning Effectiveness (Q_{LE})

Arriving at the learning effectiveness based on the 'Assessed Index' i.e. the quality of the assessment result cannot provide the extent of learning and hence there is a need to calibrate the *Assessed Index* with respect to the *Learning Index* as the benchmark. More formally, Learning Effectiveness:

$$Q_{LE} = Q_{LI} \times Q_{AI} \qquad (13)$$

4. Case Study

The proposed model is illustrated for one of the fundamental yet critical course in theory of Computer Science viz. "*Design and Analysis of Algorithms*"[2]. For the illustration of the process involved, the model has been demonstrated for one of the course objectives. However for arriving at the Learning Effectiveness for the course, sample space is given in Table 2.

Table 2: Case Study sample space

T	***N***	C_i
25	10040	177750

4.1. The Process

- *Course Objective*:

Introduce algorithm design techniques [5] namely Brute-force, Greedy, Divide-and-Conquer and Dynamic-Programming

- *Learning Outcome*:

Learner shall be able to identify and apply the appropriate algorithm design techniques for solving a problem

- *Explicit Genomes*:

Algorithm, Design Technique, Brute-force, Greedy, Divide-and-Conquer, Dynamic–Programming, Problem, Solving

- *Derived Genomes*:

Example Algorithms, Comparison of Design techniques, Problem Solving Technique

- *Classification of Genomes*:

In this study we classified the explicit and derived genomes and the competency level coefficients as per our domain knowledge and our Education Center objectives as given in Table 3.

- *Seed Questions*:

Which of the following is an example algorithm for *Brute-force* design technique?

Answer Choice 1: Selection Sort
Answer Choice 2: Quick Sort
Answer Choice 3: Generation of Fibonacci Numbers
Answer Choice 4: Travelling Salesperson Problem

- *Derived Questions*:

– Which of the following is an example algorithm for *Divide-and-Conquer* design technique?
– Which of the following is an example algorithm for *Greedy* design technique?
– Which of the following is an example algorithm for *Dynamic- Programming* design technique?
– Which of the following is an example algorithm for Divide-and-Conquer design technique?
– Which of the following is an example algorithm for Greedy design technique?
– Which of the following is an example algorithm for Dynamic- Programming design technique?
– Which of the following is NOT an *essential characteristic* for Divide-and-Conquer design technique?
– Which of the following is NOT an *essential characteristic* for Greedy design technique?

Table 3: Classification of Genomes for the case study

Category	***Genomes***	X_i	W_i	$W_i \times X_i$
Must to Know	Example Algorithms, Problem Solving Technique, Brute-force, Greedy, Divide-and-Conquer	5	1.0	5.0
Need to Know	Dynamic-Programming	1	0.75	0.75
Good to Know	Comparison of Design techniques, Problem, Solving	3	0.5	10

4.2. Findings

As per the process depicted in section 4.1 followings are the findings:

Learning Index: $Q_{LI} = 0.8055$

Assessed Index: $Q_{AI} = 0.7081$

Learning Effectiveness: $Q_{LE} = 0.5703$

The case study reveals that for one objective of the identified course "Design and Analysis of Algorithms" with a Learning Index benchmark of 0.8055, we are successful in achieving Learning Effectiveness of 0.5705.

5. Future Directions

In our model the residual seed questions were not taken into account in generating assessment because of relatively poor rank compared to rank of Core Assessment seed questions. The model can be extended to utilize the residual seed questions effectively. Moreover, Genomes can be used for arriving at pedagogy mechanisms.

6. Conclusion

The benefits of the model include:

1) The model bridges the gap between the course objectives and the assessed learning outcomes
2) Generates an outsized question bank for the assessment
3) Identify the semantics of the course objectives resulting in design of quality questions
4) Supports early validation of questions at assessment design stage
5) Projects the learning benchmark in the form of learning Index and assess the learning effectiveness

7. References

[1] Will Thalheimer (2009), A Work Learning Research Department: Aligning the learning and Performance Contexts-Creating Spontaneous Remembering.

[2] Kenneth H. Rosen (2007), Discreet Mathematics & it's Application with Combinatorics and Graph Theory).

[3] Eric P. Soulsby (2009), University of Connecticut; How to write program objectives/outcomes.

[4] B. S. Bloom (Ed.) Susan Fauer Company, Inc. 1956, Taxonomy of Educational Objectives: The Classification of Educational Goals; pp. 201–207.

[5] Cormen, Leiserson, Rivest, Stein (2001), Introduction to Algorithms, Second Edition, Prentice Hall.

[6] Anna Huang (2007), Similarity Measures for Text Document Clustering; pp. 5–6.

Using Data Mining Techniques for Inquiry Learning

C. Obimbo[1] **and Y. Zhang**[2]
[1]School of Computer Science, University of Guelph, Guelph, ON, Canada
[2]Advanced Micro Devices (AMD), Toronto, ON, Canada

Abstract - *Data mining has wide and diverse applications. In this paper we attempt to examine data mining from an educational perspective. Inquiry learning is an educational approach most commonly used by educational practitioners. Based on technologies of knowledge discovery from data mining, we present a practical approach of an inquiry learning model that aids efficient learning. The approach may be used to facilitate and enhance students learning abilities to obtain knowledge.*

Keywords: Data Mining, Inquiry Learning, Education.

1 Introduction

Inquiry learning model is widely used by educational practitioners. Inquiry learning is a process of learning through investigation, starting by that students raise questions, then they find out answers stemming from their curiosity.

In the inquiry learning model, students are considered as active participants. They are confronted with problems and expected to use their creativity and available resources to find a solution to the problems. It is an efficient learning approach especially when students are expected to study independently beyond the classroom and the school.

The use of computers has increased tremendously in science education [1, 2], especially in inquiry-based learning [3]. The Internet is an important part of inquiry-based learning as it is interactive, which helps engage students in inquiry. In addition, it can quickly provide rich sources of learning and learning materials. However, the growth of new information is at an unprecedented rate today. Huge collections of data can be easily accessed through ubiquitous computers, unfortunately, it is not an easy task for students to achieve fast results from considerable volume of existing and newly appearing data resources.

Data mining is a powerful tool for analysis of large data sets and extraction of information from it. It aims to discover useful information from large quantities of data. Data mining has been successfully used in many fields such as biomedical and DNA data analysis [4, 5], marketing and finance [6, 7, 8], and scientific databases [9, 10], etc. In this paper, we consider data mining from an education perspective, as data mining and inquiry learning have common aims: both are concerned with discovering information in data, the technology of information discovery in data mining may be used to aid students to efficiently accomplish a given inquiry-based learning task. Moreover, a little effort has been done about the relationship between human learning and data mining, although a number of articles [11, 12, 13, 14, 15] can be found about the relationship between machine learning and data mining in the literature.

We start with a discussion of inquiry learning, and then we briefly review data mining technology. A new structured and

2 Traditional Inquiry Learning

As shown in Figure 1, an inquiry learning process follows five basic stages, *Ask*, *Investigate*, *Create*, *Discuss* and *Reflect.*

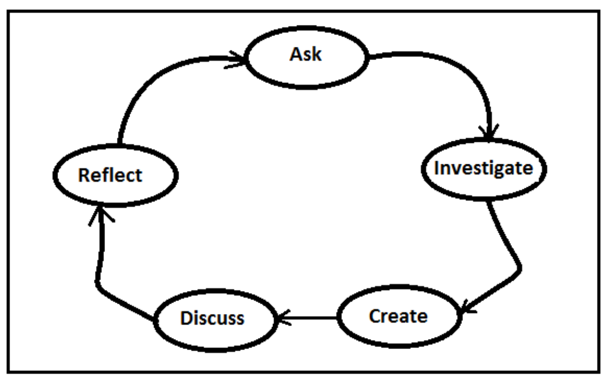


Figure 1: Inquiry learning process [16].

These stages are briefly described below [16]:

2.1 Ask

This stage usually starts with questioning. Students determine what they need to learn. The learning task is defined and described. The learning goal is often represented as a question or a problem, which will lead students to self-knowledge through aggressive inquiry in the following stages

2.2 Investigate

Focusing on the question proposed in the ask stage, students use available resources to gather information and isolate relevant variables.

2.3 Create

Students make connections between the information collected in the investigation stage. Then, they shape significant new thoughts, ideas, and theories outside their prior experience.

2.4 Discuss

After acquiring new knowledge in the "Create" stage, students need to discuss with others about their experiences and investigations, and share their new ideas with others.

2.5 Reflect

Students review the question, the research path and conclusions made. At this stage, some new questions might come into light, then a new circle of inquiry begins again. For a thorough description of the inquiry learning process, please Conclusions

3 References

Data mining evolves from a combination of multiple disciplines, including statistics, machine learning and databases. The data mining process generally involves the following major steps [17]:

3.1 Define the Problem: identify the objective

Collect and clean data: gather relevant data, transform data to a certain format, or integrate data from different sources, and enhance data quality. This stage plays an important role in the total life cycle of the whole data mining process.

3.2 Run mining algorithm:

Select an appropriate mining algorithm and apply it to the data sets.

3.3 Analyze Results

Finally evaluate the model.

As data mining evolves from combination of different disciplines, most techniques used for data mining come from other related fields. These techniques can be classified according to different views. [18] classifies the commonly used techniques into two categories, classical techniques and next generation techniques. Classical techniques includes statistics, neighborhoods and clustering, while next generation techniques includes trees, networks and rules.

Classical techniques:

- **Statistics**: Some data mining algorithms are often based on the conceptual principles and approaches of statistics.

- **Neighborhoods**: This technique can determine what a prediction value is in one record by looking for records with similar predictor values in the historical database, and using the prediction value from the record that is nearest to the unclassified record.

- **Clustering:** Clustering is the process of dividing a large number of attributes into a relatively small set of groups, or "clusters". Each cluster is highly different from other clusters, but objects within a cluster have similar characteristics. More information about Clustering can be found in [19, 20]

- **Next generation techniques:**

  - **Decision Trees:** Decision trees visually present the relationships found in data. In this model, sets of data are represented in the form of a decision tree structure. These decisions generate rules to classify the data set, and can also predict which category a new case belongs to. For a full discussion on decision trees please refer to [21].
  - **Neural networks:** Computer programs implementing sophisticated pattern detection and machine learning algorithms to build predictive models from large historical databases.
  - **Rules Induction:** The extraction of useful if-then rules from data based on statistical significance.

4 Proposed Inquiry Learning Method

From the discussion above, we have seen that in both approaches, data mining and inquiry learning, the skills for processing information are similar, and their aim is the same – to obtain knowledge. Traditional inquiry learning approach is a good tool for solving a number of problems, but may fail when dealing with a large data set. While one of the striking advantages of data mining is its ability to deal with huge data. Based on understanding of data mining and inquiry learning, we propose a new hybrid learning model, named data mining based inquiry learning model. Our attempt is to integrate data mining methodology into traditional inquiry-based learning, and provide students with a more effective strategy for systematic investigation when they experience inquiry learning processes. Given a new learning task, our proposed model processes, as can be seen in Figure 2, are as follows:

1. Topic Identification

Similar to inquiry-based learning, the first step of the proposed model is to identify the learning task, raise questions, and define the problems to be investigated. Constructing accurate and comprehensive profiles of the topic is the key issue in this part.

2. Data Gathering and Preparation

In this stage, students first collect information, then organize and refine the data, isolating relevant variables that respond to the focusing question. The information can be gathered from various sources of information, such as the Internet, books, magazines, and other technologies. Thus, students may start with using search engines on the Internet, such as Google, MSN and Yahoo, to find relevant documents by keyword indices, or create specified directories to collect relevant data.

3. Data Investigation

This stage involves data cleaning and exploration. The information collected in previous stage might include noise. Generally, erroneous data are regarded as noise and need to be filtered. Thus, during this step students need to determine what is highly relevant and what is not based upon the nature of the analytic problem. A wide range of data preparation techniques of data mining, alone or in combination, may be applied.

4. Data Mining Algorithm Applying

Based on the nature of the problem being focus and the data gathered previously, students can either apply one appropriate data mining algorithm discussed in section 2, or interact multiple algorithms to the collected data. For example, one home assignment for students in a Business Entrepreneurship class [22] is to find out an optimal new store location for a particular business. After collecting a variety of demographic data and

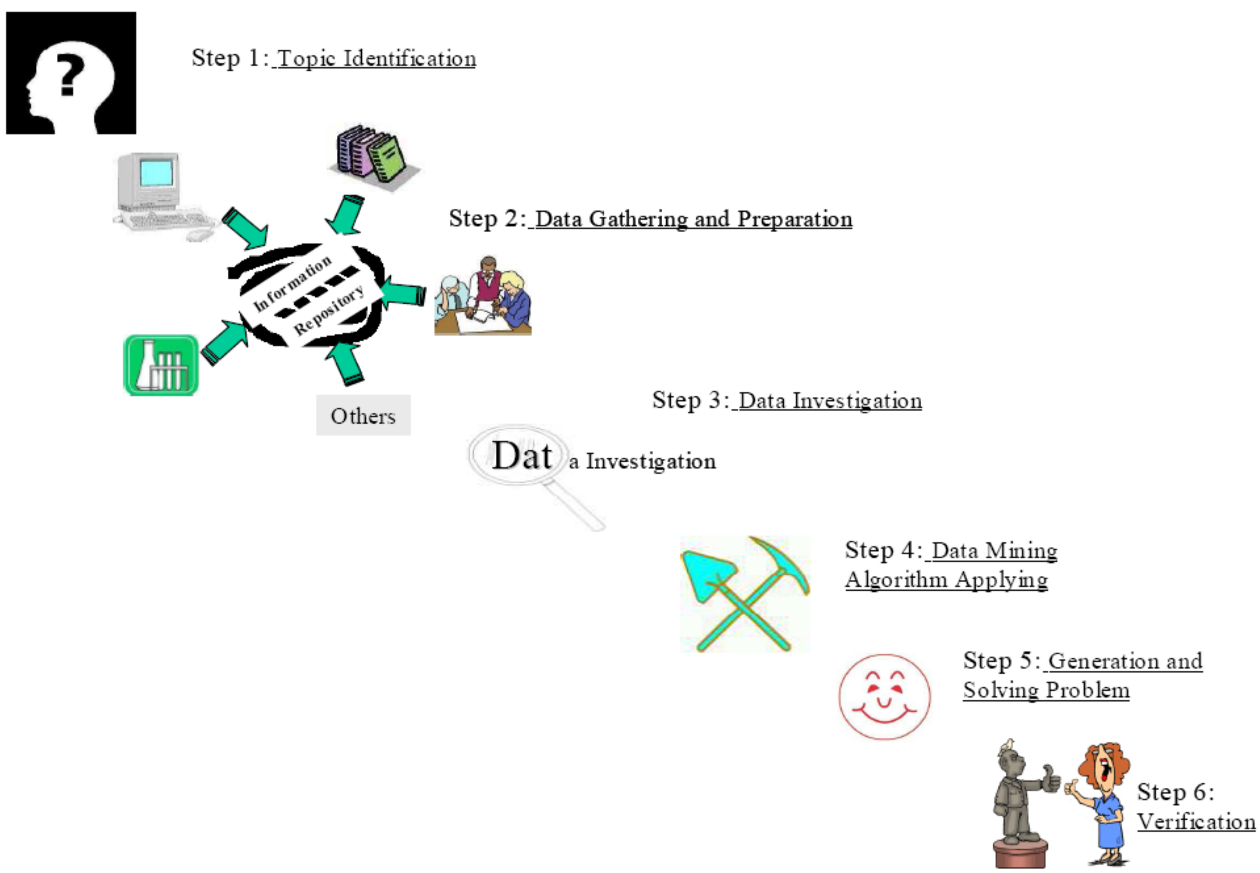


Figure 2: Proposed inquiry learning model

the buying habits of the customers, students apply Clustering techniques as the first step to identify areas that contain the majority of the customer, and then develop a list of important demographic characteristics of the population in those customer areas. Finally, they may use Decision trees or Neural networks to explore further demographic relationships between those customer areas, and identify cpotential new store locations for this business.

5. Generalization and Solving Problem

The success of this step relies upon the selection of data and analysis of data during step 2 and step 3. The goal of this step is the same as the fourth stage in traditional inquiry learning discussed in section 3. The learner undertakes the creative task of extracting information from data. He or she summarizes and draws conclusions, formulates rules or explanations.

5 References

[1] Scope of Computer Science Engineering in the future. Available from: http://www.articlesbase.com/college-and-university-articles/scope-of-computer-science-engineering-in-the-future-3582572.html, Posted: Oct 31, 2010, Viewed Mar 10, 2011

[2] G. P. Thomas. Toward effective computer use in high school science education: Where to from here? Education and Information Technologies, 6(1):29–41, 2001.

[3] M. Maxwell. Is technology in education promising too much? A neo-luddite analysis of it in curriculum. Encounter: education for meaning and social justice, 12(4):36–46, 1999.

[4] F. Wang, M. B. Kinzie, P. McGuire and E. Pan. Applying Technology to Inquiry-Based Learning in Early Childhood Education. Early Childhood Education Journal Vol 37:5, 381-389, 2010.

[5] J. P. Kubicek. Inquiry-based learning, the nature of science, and computer technology: New possibilities in science education. Canadian Journal of Learning and Technology, 31(1), 2005.

[6] S. Cho and H. Won. Data mining for gene expression profiles from dna microarray. International Journal of Software Engineering and Knowledge Engineering, 13(6):593–608, 2003.

[7] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA visual and analytic data mining. In R. Yagel and H. Hagen, editors, IEEE Visualization ’97, pages 437–442. IEEE, 1997.

[8] H. C. Jessen and G. Paliouras. Data mining in economics, finance, and marketing. In Machine Learning and Its Applications, pages 295–299, 2001.

[9] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In D. Hand, D. Keim, and R. Ng, editors, Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02), pages 61–70, New York, July 23–26 2002. ACM Press.

[10] A. Buchner and M. Mulvenna. Discovering Internet marketing intelligence through online analytical Web usage mining. SIGMOD Record (ACM Special Interest Group on Management of Data), 27(4).

[11] Wei Wang, Xiangliang Zhang, Georgios Pitsilis, “Abstracting audit data for lightweight intrusion detection”, 6th International Conference on Information Systems Security (ICISS’2010), pp. 201-215, LNCS, Springer, Gandhinagar Gujarat, India, December 15-19, 2010

[12] T. M. Mitchell. Machine learning and data mining. Commun. ACM, 42(11):30–36, 1999.

[13] B. Piwowarski and P. Gallinari. A machine learning model for information retrieval with structured documents. In P. Perner, editor, Machine Learning and Data Mining in Pattern Recognition, pages 425–438, Leipzig, Germany, July 2003. Springer Verlag.

[14] I. Bruha. Pre- and post-processing in machine learning and data mining. In Machine Learning and Its Applications, pages 258–266, 2001.

[15] N. Lavrac. Machine learning for data mining in medicine. In AIMDM, pages 47–64, 1999.

[16] University of Illinois at Urbana-chappaign. Inquiry page: Learning begins with questions. http://inquiry.uiuc.edu/index.php3, 2005.

[17] Z. Chen. Data Mining and Uncertain Reasoning: An Integrated Approach. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[18] A. Berson, S. Smith, and K. Thearling. Building Data Mining Applications for CRM. Mc-Graw Hill, New York, NY, 2000.

[19] J. Han, M. Kamber, and A. K. H. Tung. Spatial clustering methods in data mining: A survey. H. Miller and J. Han (eds.), Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2001.

[20] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[21] J. A. Berry and G. Lindoff. Data Mining Techniques. Wiley Computer Publishing, 1997.

[22] Geographic Information and Analysis at the State University of New York at Buffalo. Business entrepreneurship: New store location. http://north.ecc.edu/gis/gis projects.html, 2005.s

On Experimental Design for Diagnosing Student Learning Problems in Higher Education

A. Allinjawi[1], P. Krause[2], L. Tang[3], and H. AL-Nuaim [4]
[1]Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia
[2] Department of Computing, Surrey University, Guildford, United Kingdom
[3] Department of Computing, Surrey University, Guildford, United Kingdom
[4]Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia

Abstract - *Students often face difficulties on the basic programming courses due to several factors. Meanwhile, many papers have presented subjective assessments for diagnosing learning problems to improve the teaching of programming in higher education. In this paper, we propose an Object Oriented conceptual map model, with an approach constructed into three levels: constructing Concept Effect Propagation Table, constructing Test Item-Concept Relationships and diagnosing Student Learning Problems with Matrix Composition. Our work is a modification of the approaches of Chen and Bai, and Chu et al, by using statistical methods, rather than fuzzy sets, for our analysis. This paper includes a statistical summary that has been used with a small sample of students in King Abdulaziz University, Jeddah, Saudi Arabia, illustrating the learning problems in an Object Oriented course. The experimental results have demonstrated that this approach might be able to aid learning and teaching in an effective way.*

Keywords: Higher Education, Programming Learning Difficulties, Object Oriented Programming, Conceptual Model.

1 Introduction

A decade ago, several studies indicated that the number of students in the United States majoring in Mathematics, Science, Engineering, and Computer Science drops by 40% between the freshman and senior years [1]. In general, students in many universities often face difficulties on the basic programming courses. Programming is a fundamental part of the computer science curriculum, but it is often problematic, because it is a complicated skill to master and not an easy subject to be studied [2-3]. Therefore, the high dropout and failure rates in introductory programming courses are a widespread problem that has motivated many researchers to propose methodologies and tools to help students. It is possible that there are several reasons and factors that cause these learning and teaching problems [2]. Maybe the most important is the lack of problem solving abilities that many students show. Due to this reason, they don’t know how to program, as they don’t know how to create algorithms.

Meanwhile, many papers have proposed approaches to “improving” the teaching of computer programming in Higher Education. However, our observation is that these approaches are normally motivated by a subjective assessment of where the problems lie. It is rare to see (a) any detailed scientific analysis of where the learning difficulties lie, and (b) a statistically valid evaluation that an intervention actually does resolve a specified learning difficulty. This problem is particularly acute in the teaching of Object Oriented program where there is a wide diversity of opinions on the balance between and sequencing of the teaching of object-oriented and procedural concepts. We will focus in this paper on the development of a rigorous experimental method for diagnosing learning problems to address weaknesses with the teaching of a module.

The structure of this paper is organized as follows. In Section 2, the problems and barriers while teaching and learning Object Oriented Programming (OOP) specifically in the Kingdom of Saudi Arabia (KSA) are identified. Section 3 addresses how the use of a conceptual model might impact the effectiveness of teaching generally in higher education, and then describes our intention of using the conceptual model to diagnose the learning status of students in King Abdul-Aziz University (KAU) in Saudi Arabia. A case study is performed and the results are analysed in Section 4. In Section 5 we conclude the study and recommend the future work. Finally, references are listed in Section 6.

2 Problems and barriers in teaching and learning OOP

Students in many universities often face difficulties in the basic programming courses and especially in OOP. A study by an ITiCSE 2001 working group ("the McCracken Group") [4] established that in their introductory courses many students do not know how to program. Many students in the introductory Computer Science (CS) courses, come without adequate problem solving skills or with a poor understanding of the basic mathematical and logical tools needed for problem solving. In addition, it appears that rather than learning the basic concepts of the field, students' energies are devoted to learning syntax. Rather than learning real problem solving skills, they resort to trial and error. In addition, rather than getting the big picture of computer science, they narrow their focus to getting a specific program to run [5]. Therefore many students have a fragile grasp of both basic programming principles and the ability to perform routine programming tasks[4] .

In KSA, the widespread acceptance of CS and the fast development of CS in public, private and corporate business areas have led to new, interesting, and various opportunities for CS graduates [6]. However, the high-school curricula in KSA are lacking in CS courses, problem solving skills, and are inadequate in the English language. Typically, the majority of students that are admitted into the CS departments in high education institutions have never had any exposure to CS concepts, and CS is completely a new domain to them. In addition, in KAU CS departments, teaching object-oriented programming remains difficult, which is due to the students' lack of English language abilities, limited resources, labs and teaching assistants [7]. Moreover, CS faculties in KAU especially those teaching OO courses, face many difficulties in teaching the concepts, such as:

- The large number of concepts in OOP that students are expected to learn while having little time for adequate practice examples;
- The insufficient time for those who teach introductory programming courses, to solve problems that are large enough to demonstrate the benefits of object oriented design;
- Lack of a graphical presentational environment.

Owing to these obstacles in teaching OOP concepts, students:

- Do not fully comprehend the OO development environments;
- Lack the understanding of the OOP concepts, and have difficulties in visualizing the state spaces;
- Cannot implement some of the OO features.

Therefore, if the students generally have significant difficulties with the concepts in OOP, the question is what are the individual problems faced by students, and for which specific concepts, and which particular concepts are perceived of as being particularly difficult to grasp?

3 Methodology

In higher education, the mission of teaching and supervising students to learn well is complex and multifaceted. In fact, one of the main goals of high educational institutions is to identify ways for students to be able to think/learn effectively, and to develop critical thinking and learning skills. In addition, the researchers' intention is to improve statistically student's grasp of particular OOP concepts with which students significantly have problems. Therefore, a statistical summary has been made with a small sample of students to identify specific lack of understanding with particular concepts in OOP according to a conceptual model.

The creation of a conceptual model is a part of the process of learning rather than a manifestation of learning itself. Thus, the concept map is becoming an increasingly widely used tool in education to represent the meaningful relationships between concepts. Hawang [8] and Chen and Bai [9] propose a conceptual map model, which provides learning suggestions by analyzing the subject materials and test results. Their approach offers an overall cognition of the subject contents by identifying the key concepts of the course and the relationships between the concepts. Then the relationships between subject concepts and test items is determined by analyzing the subject materials and the item bank, and determining the learning problems of each student according to these relationships in order to diagnose the student learning status. This system can provide both objective assessments and personalized suggestions for each student by analyzing their answers and the relationships between the subject concepts and the test items. Moreover, it might be useful to help the instructors to focus on particular concepts by indentifying the key concepts that need to be covered in the course, and may help the instructors to design a particular exam related to the course which might effectively diagnose the students' specific learning problems with particular concepts.

Therefore, to diagnose the learning status of a student, a concept effect propagation approach was employed to detect student's learning problems in OO course. In OO courses, students learn new concepts and new relationships among previously learned concepts. This knowledge can be represented as a conceptual model to illustrate the relations between concepts. Figure 1 illustrates the hierarchy of related OO concepts and may produce adequate ways of measurement and assessment. These concepts are taught in OO course (CPCS203) in the first semester, year 2010/2011, in Computing and Information Technology College, female section, KAU, Jeddah, KSA. The flow of the conceptual model in figure 1 illustrates the concepts relationships, based on the course syllabus and the text books, "A Comprehensive Introduction to Object-Oriented Programming with Java" "Java How to Program: Early Objects Version" and "Java, Object-Oriented Problem Solving", and administered by the faculty member who is currently teaching the OO course. However, the design is mainly concerned with the basic concept of building classes; it did not focus on the high level of OO concepts, such as abstract and polymorphism. These concepts were not included because the faculty did not address them in the mid-term exam, and due to the complexity these concepts would add to the model if included, making it hard to demonstrate the problems faced by that students while learning OO concept. However, these concepts could be included in a separate conceptual model to diagnose the difficulties of high level of OO concepts.

The approach and model steps to diagnose the problems followed Chu's et al's model [10]. The approach constructed three levels as presented in the following details:

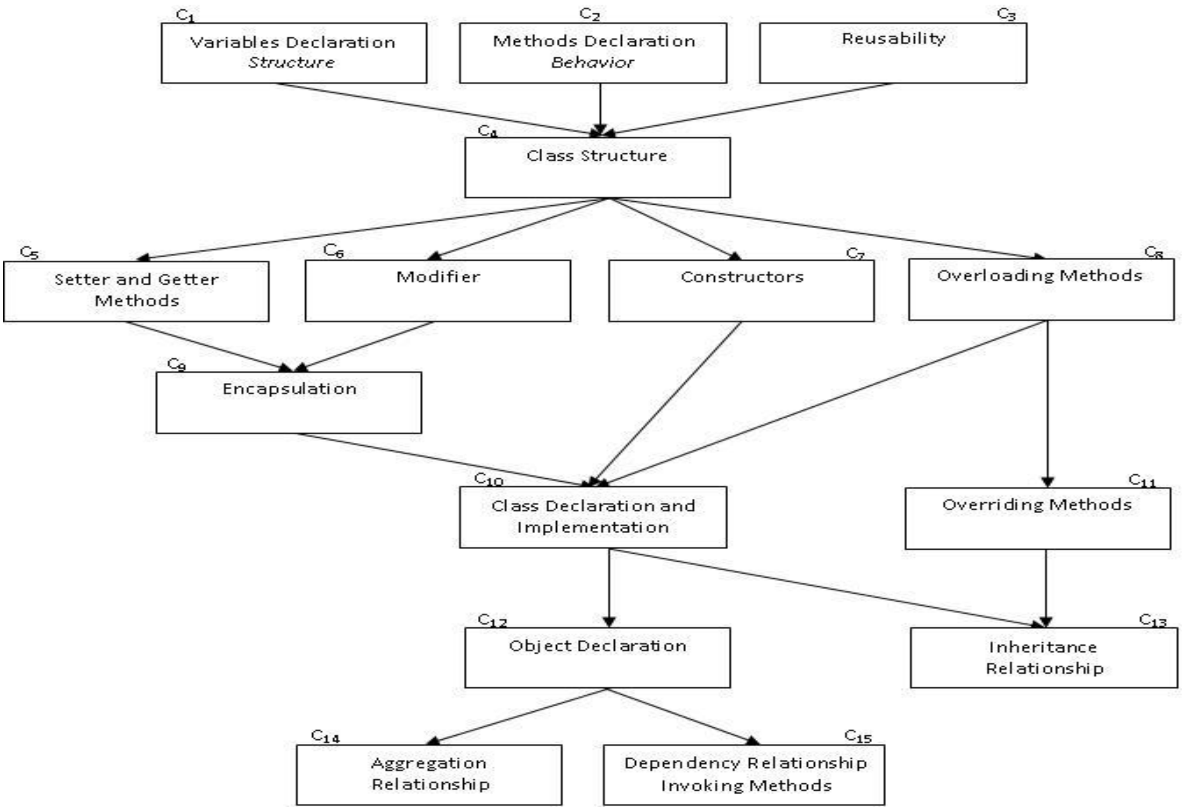


Fig. 1. Object Oriented Concept-Effect Relationship Diagram

3.1 Constructing concept effect propagation table

A *concept-effect propagation table* (CEPT) is used to model the concept-effect propagation relationships among concepts. The CEPT records all concepts that may be influenced by each concept in the student learning process. For example, in table 1, the concepts affected by C_1 are C_1, $C_4 \ldots C_{15}$. Also, the concepts affected by C_{10} are C_{12}, C_{13}, C_{14} and C_{15}. In general, C_j represents the concepts affected by C_i. If CEPT(C_i,C_j)=1, that means "C_j is one of the concepts affected by C_i during the student learning process", and if CEPT(C_i ,C_j)=0, that means "C_i is not a prerequisite concept of C_j" and student does not need C_i to learn C_j.

3.2 Constructing test item-concept relationships

A midterm exam was given to 26 computing and information technology students, three students were absent, on the 27th of December 2010. The exam consisted of three parts which examined the students' basic logical thinking in OO concepts.

Part 1 (Q_1) is about tracing a given code to generate the output. It examines the students' understanding of code flow (arrays and loops) with the inheritance structure, and how they pass parameters to the constructors and methods in the parent and child classes. However, the arrays and loops marks were not included in the test item model; the model is concerning only on OO concepts. The total marks for part 1 was 5.

Parts 2 and 3 cover most of the course concepts, which required the students to write and design complete code according to the questions' requirements. If the students answered these two parts correctly that means they understood most concepts in OOP. However, part 2 was divided into two items (Q_2 and Q_3), Q_2 is concerning on how students designed the given hierarchy model, and Q_3 is about how they declared each child class. This division will help to demonstrate the students' difficulties with specific concepts. The total marks for part 2 is 10, divided into 3.5 for Q_2 and 6.5 for Q_3.

Furthermore, part 3 was divided into two items (Q_4 and Q_5), Q_4 asks students to illustrate an aggregation relationship and declare a class with its behavior and structure. Q_5 is about how students would structure an application and declare objects to perform the requirements. The total marks of part 3 is 10, divided into 5 for Q_2 and 3.5 for Q_3. 1.5 marks to assess the loop and array structures were not included in the test item model, because structural programming was out of the study's scope.

Table 2 represents *test item relationship table* (TIRT). Each TIRT(Q_n, C_i) entry represents the degree of association between test item Q_n and concept C_i, which was calculated according to each item's (Q_n) mark. Each

Table 1. Object Oriented Concept-Effect Propagation Table (CEPT)

Prerequisite concept C_i	Propagated effect concept C_j														
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
C_1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
C_2	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
C_3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
C_4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
C_5	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1
C_6	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
C_7	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1
C_8	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1
C_9	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1
C_{10}	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1
C_{11}	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
C_{12}	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
C_{13}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
C_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
C_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 2. Object Oriented Test Item Relationship Table (TIRT)

Test item Q_n	Concept C_j														
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
Q_1	0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0.3
Q_2	0.357	0	0.857	0.286	0	0.214	0.357	0.071	0.143	0	0.071	0	0.214	0	0
Q_3	0	0.462	0	0	0.231	0	0.462	0	0.231	1	0.231	0	0.308	0	0
Q_4	0.15	0	0	0	0.3	0	0.25	0	0	0	0	0	0	0.3	0
Q_5	0	0	0	0	0	0	0	0	0	0	0	0.571	0	0	0.429

part of the solution on each item was related to specific concepts. The total mark for each concept in Q_n is represented in the TIRT(Q_n, C_i) entry. For example, Q_3 was associated with the inheritance concept by the degree 2 out of 6.5. However, since each Q_n has different mark, the degree of association range must be unified and calculated to be from 0 to *r*, where 0 indicates that the concept is not associated with the item and *r* was valued to be 1. This range was based on Chu et al's [10] proposed model. For example, as a result Q_3 was associated with the inheritance concept by the degree 0.308 out of 1.

Furthermore, it is important to recognize that the total TIRT entries of all concepts in each item (Q_n) is not essentially equal, as addressed in Chu's et al model [10]. Each TIRT(Q_n, C_i) entry is valued according to each statement "code" the students wrote while answering the questions. It could be that one statement is related to two or more concepts, which means the statement's mark must be assigned to both concepts. For example, as shown below in Q_3 when students write the constructor of an inherited class correctly, that means they understood the inheritance structure by forming the super identifier, and the constructor declaration concepts. Therefore the mark for this statement is associated to the two mentioned concepts equally.

class Triangle extends TwoDimensionalShape {

public Triangle (double Base, double VerticalHeight)
{ Super(Base, VerticalHeight); }

} *Performing the inheritance concept*

Performing the constructor declaration

3.3 Diagnosing student learning problems with matrix composition

This matrix represents the answers of the students in an *answer sheet table* (AST). Table 3 illustrates the 23 students marks who answered the midterm exam, where each entry AST(S_k, Q_n) is a value ranging from 0 to 1; 0 indicates that student S_k answered test item Q_n correctly, 1 indicates that S_k failed to answer Q_n correctly, and a value between 0 and 1 indicates a partially correct answer. The answered values were calculated to be between a unified range, as shown in the equation (1) below, because each item has different mark. The values have been subtracted from 1 to be equivalent with the 0 and 1's indication. Therefore, the value of AST(S_k, Q_n) ranged from 0 to 1, as addressed in Chu's et al model [10].

$$AST(Sk, Q_n) = 1 - ((Sk, Q_n)\ mark / the\ Q_n\ total\ mark) \quad (1)$$

Table 3. Object Oriented Answer Sheet Table (AST)

		Test item Q_n				
		Q_1	Q_2	Q_3	Q_4	Q_5
The Students S_k	S_1	0.050	0.500	0.923	0.450	0.071
	S_2	1.000	0.857	1.000	1.000	1.000
	S_3	0.300	0.571	0.462	0.400	0.786
	S_4	0.300	0.571	0.923	0.400	0.500
	S_5	0.000	0.000	0.231	0.300	0.000
	S_6	0.900	0.929	0.923	1.000	1.000
	S_7	0.050	0.286	0.231	0.300	0.000
	S_8	0.000	0.214	0.038	0.500	0.000
	S_9	0.000	0.357	0.500	0.300	0.357
	S_{10}	0.050	0.286	1.000	0.400	0.000
	S_{11}	0.200	0.929	0.231	0.700	0.071
	S_{12}	0.000	0.500	0.654	0.400	0.643
	S_{13}	0.400	0.786	0.231	0.300	1.000
	S_{14}	0.050	0.714	0.000	0.850	0.857
	S_{15}	0.300	0.571	0.923	0.350	1.000
	S_{16}	0.000	0.357	0.308	0.150	0.000
	S_{17}	0.000	0.714	0.115	0.450	0.357
	S_{18}	0.900	0.929	0.923	0.600	0.714
	S_{19}	0.000	0.357	0.231	0.300	0.000
	S_{20}	0.000	0.214	0.231	0.300	0.000
	S_{21}	0.000	0.000	0.269	0.300	0.000
	S_{22}	0.250	0.143	0.231	0.300	0.000
	S_{23}	0.050	0.500	0.500	0.300	0.000

4 Analysis and Result

After generating the three tables AST, TIRT and CEPT, there must be a relationship between the given tables to diagnose students' learning problems individually. Therefore, following Chu's et al study [10], A Max-Min composition method is used. To illustrate the Max-Min composition, let:

$$R1=\{(x,y)|(x,y)\in X\times Y\} \text{ and } R2=\{(y,z)|(y,z)\in Y\times Z\} \quad (2)$$

and the Max-Min composition will be:

$$R1 \circ R2 = \{ (x,z)|(x,z) = Max\{Min\{\mu R1(x,y), \mu R2(y,z)\}\} \text{ for } x\in X, y\in Y \text{ and } z\in Z\} \quad (3)$$

Therefore, applying the above method to the given tables can derive the error degree for each students S_k regarding each concept:

$$Error\_Degree(S_k,C_j) = AST(S_k, Q_n) \circ TIRT(Q_n,C_i) \circ CEPT(C_i, C_j) \quad (4)$$

The following is an example to illustrate the relationship:

$$AST \circ TIRT (S_1,C_1) = MAX\{MIN((S_1,Q_1), (Q_1,C_1)); MIN((S_1,Q_2), (Q_2,C_1)); MIN((S_1,Q_3), (Q_3,C_1)); MIN((S_1,Q_4), (Q_4,C_1)); MIN((S_1,Q_5), (Q_5,C_1))\} \quad (5)$$

Table 4 shows the error degree for each student with each concept after performing the above equation (4) of $Error\_Degree(S_k,C_j)$.

However, in Chu's et al [10] model a fuzzy inference was used to generate learning guidance for each student. In their work the reasons why specific membership functions were used to generate the learning guidance was not made sufficiently clear, and we are not confident in the validity of their use. Therefore, in our case study a statistical summary has been used to identify the lack of understanding within these particular concepts in OOP. By generating the mean, median and the standard deviation of the error degrees for each concept, we obtained with significant confidence results that demonstrate in which particular concepts students are facing problems while learning. From table 5, it is shown that most of the students have problems with class declaration and implementation, object declaration, inheritance relationships, aggregation relationships and dependency relationship concepts. This means students might understand the prerequisite concepts of the declaration of classes, but getting the big picture, ongoing in depth on how to demonstrate the relationship between classes and how to structure a high level application that performs particular requirements, is difficult.

5 Conclusion and Future work

To diagnose students learning problems with a specific course is an important research topic of adaptive learning systems. In this paper, we have presented a model for identifying learning problems of students while learning OOP concepts in KAU. The result illustrates that students do not demonstrate a strong ability to effectively use the relationships between classes to design and structure a high level application that meets particular requirements with an OO environment. However, more work needs to be done to clearly diagnose these problems which might be due to, the exam structure, students not grasping the abstract concepts or due to the instructors' teaching style that may not matching the students' needs.

We believe identifying the key concepts and their relations between them in different fields to create a conceptual map model might be useful. It may help the instructors to design a particular exam that might effectively diagnose the students' learning problems with any given concepts. Through the identification of such problems, instructors can adjust their instruction to the students to improve their learning process.

We have illustrated the approach with a small sample of students. This of course leaves us with a low confidence in the results' consistency. Our next step will be to repeat the same experiment with a richer statistical and assessment method supported by a larger number of students, in the next semester. These statistical assessment methods will measure the students' performance, skills, knowledge and their ability to comprehend the same particular concepts, and to identify

Table 4. The Error Degree table

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
S_1	0.357	0.462	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.923	0.500	0.923	0.923	0.923	0.923
S_2	0.357	0.462	0.857	0.857	0.857	0.857	0.857	0.857	0.857	1.000	0.857	1.000	1.000	1.000	1.000
S_3	0.357	0.462	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571
S_4	0.357	0.462	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.923	0.571	0.923	0.923	0.923	0.923
S_5	0.150	0.231	0.000	0.231	0.300	0.231	0.250	0.231	0.300	0.300	0.231	0.300	0.300	0.300	0.300
S_6	0.357	0.462	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.923	0.857	0.923	0.923	0.923	0.923
S_7	0.286	0.231	0.286	0.286	0.300	0.286	0.286	0.286	0.300	0.300	0.286	0.300	0.300	0.300	0.300
S_8	0.214	0.038	0.214	0.214	0.300	0.214	0.250	0.214	0.300	0.300	0.214	0.300	0.300	0.300	0.300
S_9	0.357	0.462	0.357	0.462	0.462	0.462	0.462	0.462	0.462	0.500	0.462	0.500	0.500	0.500	0.500
S_{10}	0.286	0.462	0.286	0.462	0.462	0.462	0.462	0.462	0.462	1.000	0.462	1.000	1.000	1.000	1.000
S_{11}	0.357	0.231	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857
S_{12}	0.357	0.462	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.654	0.500	0.654	0.654	0.654	0.654
S_{13}	0.357	0.231	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786
S_{14}	0.357	0.000	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714
S_{15}	0.357	0.462	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.923	0.571	0.923	0.923	0.923	0.923
S_{16}	0.357	0.308	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357
S_{17}	0.357	0.115	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714
S_{18}	0.357	0.462	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.923	0.857	0.923	0.923	0.923	0.923
S_{19}	0.357	0.231	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357
S_{20}	0.214	0.231	0.214	0.231	0.300	0.231	0.250	0.231	0.300	0.300	0.231	0.300	0.300	0.300	0.300
S_{21}	0.150	0.269	0.000	0.269	0.300	0.269	0.269	0.269	0.300	0.300	0.269	0.300	0.300	0.300	0.300
S_{22}	0.150	0.231	0.143	0.231	0.300	0.231	0.250	0.231	0.300	0.300	0.231	0.300	0.300	0.300	0.300
S_{23}	0.357	0.462	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500

Table 5. The statistical summary of the Error Degrees of each concept

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
Mean	.31139	.32300	.48126	.51978	.53448	.51978	.52390	.51978	.53448	.62717	.51978	.62717	.62717	.62717	.62717
Median	.35700	.30800	.50000	.50000	.50000	.50000	.50000	.50000	.50000	.65400	.50000	.65400	.65400	.65400	.65400
Std. Deviation	.077367	.151318	.272548	.226749	.208321	.226749	.221391	.226749	.208321	.272460	.226749	.272460	.272460	.272460	.272460

whether these students understand the OO relationship concepts or not. Moreover, according to the debate and the case studies for engaging visualization tools with the educational aspects, such as the educators' confidence and persistence of the advantages of using visual tools while learning OOP concepts, and after statistically defining the students understanding barriers with the OO relationship concepts, it is challenging to assess again, using a quantitative analysis, another group of students engaging with the proposed teaching methods (visualization tools) within particular concepts (inheritance, aggregation and dependency), and assess whether that change will make a difference in the students learning outcomes.

6 References

[1] F. P. Deek, H. Kimmel, and J.A. McHugh, "Pedagogical changes in the delivery of the first-course in computer science: Problem solving, then programming," *Journal of Engineering Education*, vol. 87, pp. 313-320, 1998.

[2] E. Lahtinen, Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," SIGCSE Bull., vol. 37, pp. 14-18, 2005.

[3] T. Jenkins, "On the Difficulty of Learning to Program," presented at the Learning and Teaching Support network , 2002.

[4] R. Lister, E.S. Adams, S. Fitzgerald, W. Fone , J. Hamer, M. Lindholm, R. McCartneyet, J. E. Moström, K. Sanders, O. Seppälä, B. Simon and L. Thomas, "A multi-national study of reading and tracing skills in novice programmers," SIGCSE Bull., vol. 36, pp. 119-150, 2004.

[5] V. H. Allan and M. V. Kolesar, "Teaching computer science: a problem solving approach that works," SIGCUE Outlook, vol. 25, pp. 2-10, 1997.

[6] A. S. Al-Salman and J. Adeniyi, "Computer science education in a Saudi Arabian university: a comparative study of its B.Sc. program," SIGCSE Bull., vol. 32, pp. 34-39, 2000.

[7] A. Allinjawi, "The Use of ALICE as a 3D Interactive Programming Environment to Teach Novice Programmers Concepts in OOP," Master, Computer Science Department, King Abdulaziz University, Jeddah, 2007.

[8] G.-J. Hwang, "A conceptual map model for developing intelligent tutoring systems," Computers & Education, vol. 40 pp. 217-235, 2003.

[9] S.-M. Chen and S.-M. Bai, "Learning barriers diagnosis based on fuzzy rules for adaptive learning systems," Expert Syst. Appl., vol. 36, pp. 11211-11220, 2009.

[10] H.-C. Chu, G. Hwang, J. C. R. Tseng, and G. Hwang, "A Computerized Approach to Diagnosing Student Learning Problems in Health Education," Asian Journal of Health and Information Sciences, vol. 1, pp. 43 - 60, 2006.

.

SESSION

TOOLS AND SYSTEMS + WEB USAGE METHODS + LEARNING ENVIRONMENTS

Chair(s)

TBA

A pedagogical agent which incorporates a text mining tool to promote collaborative writing

Isis da Costa Pinho[1], Eliseo Reategui[1], Alexandre Moretto Ribeiro[1], Mateus David Finco[2]

[1]PPGIE / PPGEDU, Federal University of Rio Grande do Sul (UFRGS)
Av. Paulo Gama, 110 - prédio 12105 - 90040-060, Porto Alegre, RS - Brazil
isis.letras@yahoo.com.br, eliseoreategui@gmail.com, alexandremorettoribeiro@gmail.com
[2]Faculdade da Serra Gaúcha
Rua os Dezoito do Forte, 2366 - 95020-472, Caxias do Sul, RS - Brazil
mateusfinco@yahoo.com.br

Abstract - *This paper discusses how mediation strategies can be used to design a pedagogical agent integrated to a text mining tool to promote collaborative work in academic reading and writing activities. The agent uses the Artificial Intelligence Markup Language (AIML) to communicate with students, while its text mining features enables it to keep track of what is being produced collaboratively. Mediation strategies based on sociocultural theory are also employed in order to endow the pedagogical agent with the capacity to intervene in students' discussions according to precise goals. A specific architecture is proposed in the paper as a form to integrate these different mechanisms in the pedagogical agent. The proposed solution explores the roles of interaction, mediation and assistance as key elements for structuring the pedagogical agent's participation as a peer guide in collaborative text writing. Keywords: pedagogical agent, text mining, collaborative writing, mediation, distance learning.*

Keywords: pedagogical agent, text mining, collaborative writing, mediation strategies

1. Introduction

The use of technology in education challenges teachers to use the Web 2.0, information and communication technologies, as well as artificial intelligence for engaging learners in intense interaction to promote collaborative and exploratory learning. Moreover, distance education has required the creation of pedagogical tools that provide adaptive assistance to students in an individualized manner to solve problems without the constant teacher supervision. One of these tools is pedagogical agents. In a time where distant teachers face a heavy workload dealing with a large number of students, pedagogical agents can serve as entities to support learning and mediation, minimizing the absence of immediate feedback from the teacher when this cannot be done.

This research aims to examine how mediation strategies may be used to structure the knowledge base of a pedagogical agent in order to support a collaborative writing process. While mediation strategies give the agent some pedagogical knowledge on how to promote discussion and to keep students engaged in their collaborative work, a text mining tool is employed in order to give the agent some domain knowledge about the topic students have to write about.

This paper is organized as follows. The next section presents pedagogical agents and proposes them as a means to mediate collaborative work. Section 3 presents theoretical background on the topics interaction and mediation. Section 4 proposes a specific architecture for pedagogical agents, based on the integration of mediation knowledge, conversational knowledge and domain specific knowledge. Section 5 discusses the pedagogical agent architecture and proposes directions for future work.

2. Pedagogical Agents

André and Rist [1] defined virtual characters as anthropomorphic computerized beings whose main goal is to assist people in performing specific tasks. Based on this definition, we describe pedagogical agents as virtual characters whose main goal is to help students in learning-related activities. They may act as virtual tutors, mentors, experts, virtual partners, providing a more natural interaction with digital educational artifacts. Pedagogical agents may also be classified into goal-driven (tutors, mentors and assistants), and utility-driven agents that perform supplementary tasks related to teaching activities [2]. In any of these cases, pedagogical agents provide an opportunity to improve the communication among students and computers by using paralinguistic cues, such as gestures and facial expressions, which are familiar to human beings. This form of human-computer communication also has the potential to establish emotional and social bonds between pedagogical agents and students, which could consequently facilitate learning [3].

The use of virtual characters as a communication tool in computing systems has spread in a wide range of applications [4]. In the educational area, Shaw & Johnson [5] showed how virtual teachers could guide students in online interactive activities. Rist and Muller [6] demonstrated that the presence of virtual characters in educational material could influence students' perceptions, making them consider topics significantly less difficult and more entertaining. A significant portion of these investigation results are aligned with the ideas that teaching and learning are highly social activities. Based on this premise, Kim and Baylor [7] showed how virtual characters could be used not as tutors, but as learning companions.

Pedagogical agents are usually designed to present information, to encourage students, to give pedagogical feedback, to guide and help learners [7]. Thus, pedagogical agents can provide a situated social interaction, playing the role of a more expert peer and addressing issues which are relevant to learners' needs. The sense of presence provided by the agents also contributes to fill out the void caused by the lag produced by distance education environments [7].

In order to promote a good interaction with users, pedagogical agents need to be credible, or appear to have life and consistency in their actions. This life-like identity is built from their autonomy, social skills and adaptability features. For that, agents should have a broad and diverse repertoire of behaviors according to likely situations, and they should seek to help students to keep their focus on relevant aspects of the learning material.

By perceiving pedagogical agents as entities which have an identity and life (personae effect), students may get more motivated to participate in activities, feeling they can be assisted at any time when needed [8][9].

Based on social-cognitive principles, Kim and Baylor [7] discuss the instructional potential of pedagogical agents as learning companions to support interaction in pairs. The authors state that, based on the ZPD metaphor, the pedagogical agent can be designed to be in a higher intellectual stage scaffolding learners' performance to advance in their knowledge construction.

Here, our main goal has been to design a pedagogical agent with the capability of mediating the communication between two students in order to get them to work together collaboratively in the best possible way. As our focus has been on tasks related to text writing, the initial goal of the agent is to support students' reading and discussing, helping them keep their focus.

Our agent uses the use the Artificial Intelligence Markup Language (AIML) associated with a text mining tool in order to communicate with students and keep track of what is being produced collaboratively. AIML employs a stimulus-response method by which stimuli (sentences and fragments) are stored and used to search for pre-defined replies [10]. The language has certain features which distinguish it from a simple retrieval of questions and answers from a database. For example, it may keep the context of a speech in order to enable the character to remember a previous statement. It also enables the launching of particular programs when a certain pattern is found. Such a feature is used in our pedagogical agent architecture to get the character to combine a previously compiled AIML knowledge base with a more dynamic representation of specific domain knowledge extracted from texts with the help of the text mining tool.

The use of an AIML knowledge base is founded on the fact that the agent has to be able to talk to students about general topics in order to be more credible and to create a social bond with them. The next section discusses the topic mediation and arguments about the importance of such strategies for a pedagogical agent.

3. Mediation Strategies

The theoretical framework that guided this research comprises the works of Vygotsky, including sociocultural theory applied to language acquisition, mediation strategies and learning processes [11][12][13][14][15]. Collaboration is known to be one of the cornerstones of educational experience. In a computer mediated setting, it may promote the creation of an online learning community through the grouping and the involvement of students. Moreover, collaboration can promote the development of abilities for reasoning in a critical way, and it can produce the co-construction of knowledge and meaning. It may also produce reflection, and trigger transformative learning [16].

According to the Vygotskian sociocultural theory, interaction and collaboration are essential to the co-construction of knowledge, as well as the development of greater control over oneself and one's own learning [13][17][14][18]. Interaction is, thus, an important factor in learning processes. In distance education, it may refer to student-student, student-tutor, student-teacher [15], system-student and student-pedagogical agent contact. However, it is the interactive nature of collaboration, with the establishment of intersubjectivity based on collective goal-directed actions, which best fosters learning. Thus, the engagement of learners in the negotiation of ideas, knowledge and perspectives for the common benefit can lead to opportunities for focusing on the sense and form of language, scaffolding hypotheses and strategy testing, creating and applying learning strategies [19][20][21].

Learning in a virtual environment is the result of the relationships between several factors including: teachers and learners' characteristics, needs, expectations and objectives, amount and type of input received, amount and type of output expected, teaching and learning styles and strategies. Furthermore, the learning process can be seen as a complex system as it is the combination of these multiple factors [22]. Learning remains open to the influence of initial and external conditions, such as feedback, task procedures, and the virtual learning environment in use.

The focus of our work is neither on the student nor on the teacher, but on the interactive flow which structures and keeps virtual communication between people. In this context, learning is socially situated, being the product of the collaboration between learners and their interaction with technology [12][17][14][18]. This process leads to the expansion of a learning zone with the potential to be attained, called Zone of Proximal Development (ZPD). It's in the zone of proximal development that pedagogical intervention and assistance of a more capable partner can produce more effects on learning. This type of assistance is called by Wood et. al. [21] scaffolding, and it consists of an interactive process of negotiation, in which the expert, based on the learner competence level, chooses the type of assistance required. Little by little, as the learner's competence evolves, the responsibility to complete the tasks has be given to him/her [21][23][24].

This process of assistance is characterized by six functions: (1) recruiting partner's attention to the task, making the partner concerned; (2) reducing the degree of freedom by simplifying or limiting the demands of the task to make it feasible; (3) maintaining the direction by keeping the focus and progress towards their goals; (4) pointing out relevant features through feedback; (5) controlling frustration by decreasing the stress of the novice; (6) and demonstrating the expected procedures to achieve the goals [21].
Therefore, knowledge acquisition can be promoted through a mediation process intentionally organized by the teacher who creates problematic situations that challenge learners to develop self-regulation. Here, these strategies have been implemented in the pedagogical agent who provides assistance to the students in their tasks.

During the social cognitive development of learners, mediation is central to the search for a greater control over one's own behavior. Through interaction, individuals may become skilled with tools; they may get acquainted with signs, meanings and knowledge. Based on this, teachers should consider in their planning and educational intervention not only material resources, but foremost, mediation strategies which are adapted to the students' profile, needs and expectations [25]. A pedagogical agent whose main goal is to assist students in developing collaborative work should also take such factors into account.

4. The Pedagogical Agent Architecture

The main goal of the pedagogical agent proposed in this work is to mediate the interaction between students who are working together in a writing activity. Fig. 1 shows a diagram in which we can see two students interacting in a collaborative work, and interacting as well with the pedagogical agent.

The agent is composed of 3 knowledge bases. One of them stores conversational knowledge, which enables the agent to talk about general topics which may interest students and which may be important for establishing a social bond with them.

The second knowledge base of the agent is composed of knowledge about mediation strategies, guiding the agent on how to intervene on students' discussions as to keep them engaged in the activity and keep their focus on the theme proposed for their assignment.

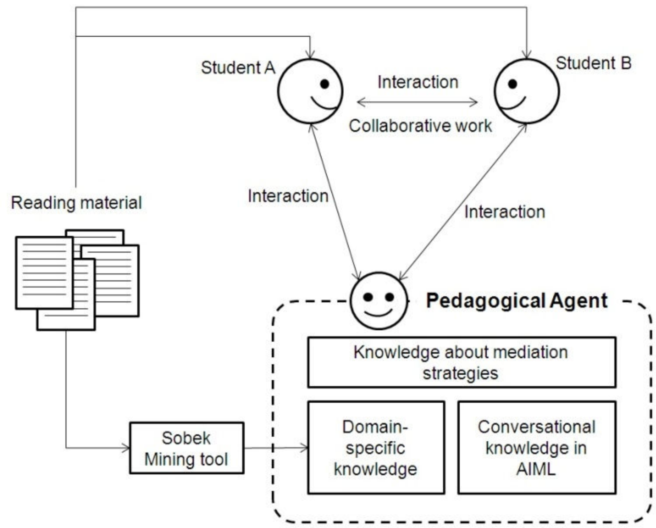


Fig. 1. architecture of the pedagogical agent.

The third knowledge base stores specific knowledge about a topic that is being discussed by the students. A set of texts about the topic feeds a text mining system. This system extracts from the texts relevant terms and relationships among them, representing this knowledge in the form of a graph. Fig. 2 shows a graph extracted from Wikipedia text about data structures1.

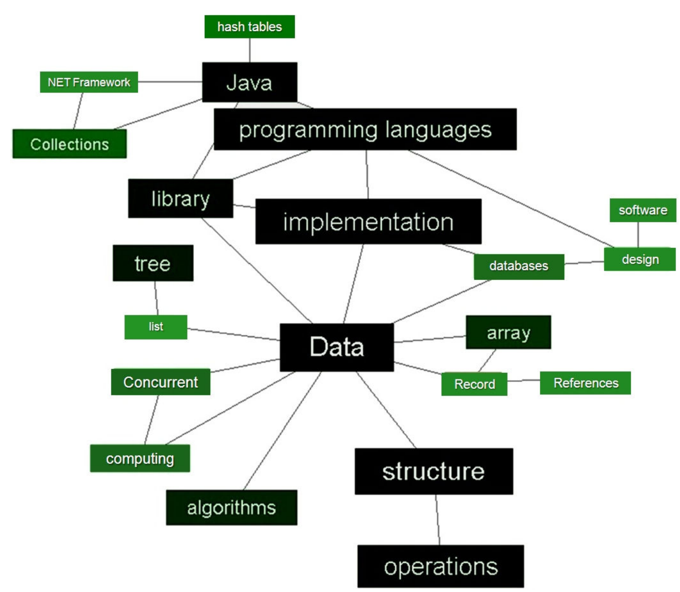


Fig. 2. graph obtained from a text on data structures.

The text mining process developed in this work has been based on Schenker's use of the n-simple distance graph model [26], a graph that has in its nodes words representing the

[1] Text extracted from: http://en.wikipedia.org/wiki/Data_structure, Accessed in March, 2011

concepts found in the text. The edges used to link nodes represent adjacent information, i.e. they represent how the concepts appear together in the text.

The text mining tool presented here has already been used in different educational applications, as in Macedo et al. [27] where the authors showed how the graphs could assist teachers in identifying certain pattern of problems in students' writings. Azevedo et al. [28], on the other hand, proposed a method for identifying the quality of contributions in discussion forums through a computational method employing the graphs extracted from the students' posts. Here, the graphs guide the pedagogical agent on how to arbitrate students' discussions with the main goal of keeping them focused on relevant concepts which have to be considered in their work.

5. Discussion and Final Considerations

The results of this study are intended to contribute to the research on the uses of computers in education, focusing on the impact of pedagogical agents in collaborative writing.

In some terms, our proposal is related to that of Macedo et al. [27], who claim that although the graphs obtained in the text mining process cannot be used to reconstruct an original text, they can provide a good understanding about ideas and key concepts. In Macedo's work, it has also been shown that the graphs elicited from students' writings can show certain features which may help teachers evaluate the quality of the students' work.

Klemann et al. [29] suggest the use of Sobek as a resource for helping students in writing activities. In the proposed methodology, the reflection on what has been read in a text, and the evaluation of the relevance of concepts and their relationships in the graphs can be of great use for organizing ideas and deepening the understanding about a topic being studied. Here, the graphs are not used directly by the students, but they are employed by the pedagogical agent as a source of knowledge for understanding which are the relevant topics which have to be covered in the students' assignment.

Although most of the work in the area has focused on positive aspects of pedagogical agents from a learning perspective, some criticism has also been presented. Gulz, for instance, discusses and questions the types of gain that can be obtained by pedagogical agents [30]. Choi & Clark demonstrated that there was no difference in students' performances when comparing the use of a pedagogical agent with a simple arrow and voice narration [31]. The authors argued that there was no evidence in favor of the pedagogical agents regarding their ability to motivate, interest or tutor the students. Here, however, we are not only concerned with the type of media used to present information to students. We are interested in the actual capacity of the agent to mediate students' interaction through the use of different mechanisms such as an AIML engine, a text mining tool and a mediation knowledge base. In this regard, our position is aligned with that of Veletsianos [32] who argued that Choi & Clark [31] disregarded important aspects of agents' properties in their research.

Regarding the design of the pedagogical agent, it is intended that the agent makes the learning environment more interactive as to provide scaffolding in students' learning process. According to the classification proposed by Giraffa and Viccari [2], our pedagogical agent can be classified as a goal-driven agent, as it is intended to work as an assistant, but it also has features of utility-driven agents, as it is in charge of the specific task of promoting the collaboration between students in writing. It is currently being integrated in a portal in the health area, with the intent to provide medical and nursing students with further support for their writing assignments. A next step in our research will be the integration of domain specific vocabulary in the agent's architecture as a means to associate more semantics to the analyses carried out by the agent.

Acknowledgment

This work has been partially supported by the National Council for Scientific and Technological Development (CNPq - Brazil) under grant 476398/2010-0, FAPERGS Research Support Foundation, under grant 1018248, and Fiocruz-Fiotec project no. ENSP 060 LIV 09.

References

[1] E. André and T. Rist. "From adaptive hypertext to personalized web companions". Communications of the ACM, vol. 45, no. 5, May 2002.

[2] L. Giraffa and R. Viccari. "The Use of Agents Techniques on Intelligent Tutoring Systems". In Proceedings of the XVIII International Conference of the Chilean Computer Science Society (SCCC '98). IEEE Computer Society, Washington, DC, USA, p.76-, 1998.

[3] N. C. Krämer and G. Bente. "Personalizing e-Learning: The social effects of pedagogical agents". Educational Psychology Review, vol. 22, no. 1, p.71-87, 2010.

[4] I. S. Pandzic. "Life on the Web". Software Focus Journal. Wiley, NY, p.52-58, 2001.

[5] E. Shaw and W. L. Johnson. "Pedagogical agents on the Web". In Proceedings of the Third International Conference on Autonomous Agents, ACM Press, NY, p.283-290, 1999.

[6] E. André, T. Rist, and J Muller. "Employing AI methods to control the behavior of animated interface agents". Applied Artificial Intelligence, v13, no. 4-5, p.415-448, May 1999.

[7] Y. Kim and A. L. Baylor. "A social-cognitive framework for pedagogical agents as learning companions". ETR&D, vol. 54, no. 6, p.569-590, 2006.

[8] T. Erickson. "Designing agents as if people mattered". In J. M. Bradshaw (Ed.), Software agents, Menlo Park, CA: MIT Press, p. 79-96, 1997.

[9] J. C. Lester, S. A. Converse, S. E. Kahler, S. T. Barlow, B. A., Stone, and R. S. Bhoga. "The persona effect: Affective impact of animated pedagogical agents". In Proceedings of Human Factors in Computing Systems (CHI ´97), Atlanta, GA, ACM Press, p.359-366, 1997.

[10] R. Wallace. The elements of AIML style. San Francisco, CA: ALICE A.I. Foundation. 2003.

[11] L. S. Vygotsky. Mind in Society: the Development of Higher Psychological Processes. Cambridge, MA.: Harvard University Press, 1978.

[12] L. S. Vygotsky. Thought and Language. Cambridge, MA: MIT Press, 1986.

[13] L. S. Vygotsky. The Collected Works of L. S. Vygotsky. V.1. Thinking and Speaking. New York, N.Y.: Plenum Press, 1987.

[14] J. P. Lantolf. "Introducing sociocultural theory". In: J. Lantolf. (Ed.), Sociocultural theory and second language learning. Oxford: Oxford University Press, p. 27-50, 2000.

[15] J. P. Lantolf. "Sociocultural Theory and L2: State of the Art". SSLA, vol. 28, p. 67-109, 2006.

[16] R. M. Palloff, K. Pratt. Collaborating online: Learning Together in Community. São Francisco, CA: Jossey-Bass, 2005.

[17] A. S. Ohta. "Rethinking Interaction in SLA: Developmentally Appropriate Assistance in the Zone of Proximal Development and the Acquisition of L2 Grammar". In J. Lantolf (Ed.), Sociocultural Theory and Second Language Learning. Oxford: Oxford University Press, p. 51-78, 2000.

[18] M. Swain. "Integrating Language and Content Teaching Through Collaborative Tasks". The Canadian Modern Language Review, vol .58, no. 1, p.44-63, Set. 2001.

[19] M. Swain. "The Output Hypothesis and Beyond: Mediating Acquisition Through Collaborative Dialogue". In: J. Lantolf (Ed.), Sociocultural Theory and Second Language Learning. Oxford: Oxford University Press p. 97-114, 2000.

[20] M. Swain, S. Lapkin. "Focus on Form Through Collaborative Dialogue: Exploring Task Effects". In: M. Bygate, P. Skehan, M. Swain (Eds.). Researching Pedagogic Tasks: Second Language Learning, Teaching and Testing. London, UK: Longman. Cap. 5, p.98-118, 2001.

[21] D. Wood, J. S. Bruner, G. Ross. "The role of tutoring in problem solving". In: Journal of Child Psychology and Child Psychiatry, vol. 17, p.89-100. 1976.

[22] D. Larsen-Freeman, L. Cameron. Complex Systems and Applied Linguistics. Oxford: Oxford University Press, 2008. 287p.

[23] L. E. Berk. "Children's Private Speech: An Overview of Theory and the Status of Research". In: R. M. Diaz, L. E. Berk (Eds.). Private Speech: From Social Interaction to Self-Regulation. New Jersey: Lawrence Erlbaum, p. 17-55, 1992.

[24] J. K. Hall. Methods for Teaching Foreign Language: Creating a Community of Learners in the Classroom. Ohio: Merrill Prentice Hall, 2001.

[25] P. A. Jaques, A. F. Andrade, J. L. Jung, R. Bordini, and R. M. Viccari. "Using Pedagogical Agents to Support Collaborative Distance Learning". In Proceedings of Computer Supported Collaborative Learning, New Jersey: Lawrence Erlbaum, p. 560-561, 2002.

[26] A. Schenker. Graph-Theoretic Techniques for Web Content Mining. PhD thesis, University of South Florida, 2003.

[27] A. Macedo, E. B. Reategui, A. Lorenzatti, P. Behar. "Using Text-Mining to Support the Evaluation of Texts Produced Collaboratively". Education and Technology for a Better World: Selected papers of the 9th World Conference on Computers in Education, Berlin: Springer, 2009.

[28] B. Azevedo, E. B. Reategui, P. Behar. "Qualitative Analysis of Discussion Forums". In: Proceedings of the IADIS International Conference on e-Learning, Freiburg, Germany, 2010.

[29] M. Klemann, A. Lorenzatti, E. B. Reategui. "O Emprego da Ferramenta de Mineração de Textos SOBEK como Apoio à Produção Textual". In Proceedings of Simpósio Brasileiro de Informática na Educação, Florianópolis, Brazil, 2009.

[30] A. Gulz. "Benefits of virtual characters in computer-based learning environments: Claims and evidence". International Journal of Artificial Intelligence in Education, vol. 14, no. 3, p.313-334, 2004.

[31] S. Choi, R. E. Clark. "Cognitive and Affective Benefits of an Animated Pedagogical Agent for Learning English as a Second Language". Journal of Educational Computing Research, vol. 34, no. 4, p.441-466, 2006.

[32] G. Veletsianos. "Cognitive and Affective Benefits of an Animated Pedagogical Agent: Considering Contextual Relevance and Aesthetics". Journal of Educational Computing Research, vol. 36, no. 4, 373-377, 2007.

Stanford Mobile Inquiry-based Learning Environment (SMILE): using mobile phones to promote student inquires in the elementary classroom

Sunmi Seol[1], Aaron Sharp[2], and Paul Kim[2]
[1]Seeds of Empowerment, Palo Alto, California, USA
[2]School of Education, Stanford University, Palo Alto, California, USA

Abstract - *This paper introduces the Stanford Mobile Inquiry-based Learning Environment (SMILE), which was designed to promote student-created questions in an engaging way in the elementary classroom using mobile phones. With SMILE, students use a mobile phone application to create questions, which are then solved and rated by peers. The entire process is controlled and monitored by a teacher with an activity management application. In a pilot study, 32 fourth- and fifth-grade students participated in SMILE and completed a subsequent survey. Students indicated that they were very satisfied with the activity because it allowed them to create their own questions and share them with peers. The student-created questions exhibited multiple dimensions of learning and were categorized into three question types according to Bloom's taxonomy: Knowledge, Comprehension and Analysis.*

Keywords: Inquiry-based learning, student-created questions, mobile phone, engaging way, peer-assessment, ad-hoc network

1 Introduction

It is generally acknowledged that student-created questions play an important role in the learning process ([8], [10], [15]) and they have been demonstrated to improve student learning outcomes ([1], [5], [23], [29]). In posing questions themselves, students must revisit previous learning materials and reshape their thoughts relating to prior learning, thereby deepening their understanding ([17], [30], [32]). Moreover, if students are made aware that they will be asked to create questions at a later time, they will actively monitor and attend to what they are learning during class in anticipation ([20], [28]). This practice is in line with Constructivist theories of learning, which presuppose that students learn best when discovering and unpacking content for themselves ([4], [21], [31]). Despite these findings, though, student-created questions have remained consistently absent from the majority of teachers' repertoires [13]. Studies have reliably shown that only a very small portion of questions asked in a classroom are created by the students ([6], [9]), implying that a powerful pedagogical tool is being underutilized.

The affordances of mobile phones present a unique opportunity to reintegrate student-created questions into the classroom. More specifically, students are already actively trying to communicate with each other during class on their mobile phones ([11], [14], [33], [34]), so there is an opportunity to redirect this communication toward class material through student-created questions. Indeed, it is slowly being recognized and demonstrated that mobile phones are highly engaging tools to be taken advantage of, not prohibited [16]. For example, data collected from four elementary and two middle school classes indicated that the use of mobile phones in the classroom increased student motivation, improving their quality of work [25]. Considering the current trend toward the consolidation of open-source mobile operating system platforms [24] and that mobile phone ownership among children has increased by 68% in the past five years [19], there could be no better time to take advantage of these affordances.

This paper introduces the Stanford Mobile Inquiry-based Learning Environment (SMILE), which enables students to quickly create multiple choice questions and share them with peers using mobile phones during class. Students can then respond to and rate each other's questions, after which they can view detailed personal and question-related data (including which student answered the most questions accurately and which student created the highest-rated question). SMILE also includes an activity management application for the teacher that allows him or her to control the progress of the activity in real time and to view all student data.

There are several important features of SMILE that were deliberately designed to maximize its effectiveness. First, allowing students to include photographs in their questions garners the learning benefits associated with presenting materials in multimedia ([18], [35]). Second, having students create multiple choice questions requires them to think critically in order to create three distracters for each question ([26], [36]). Third, permitting students to rate each other's questions provides feedback and incorporates an element of peer-assessment, which has been demonstrated to be valuable to a majority of students ([27], [37]). Fourth, allowing students to view who scored the highest may foster a "non-pressured" competitive learning environment [21], which has been demonstrated to increase intrinsic motivation. Finally, supplying the teacher with all of the students' questions and responses through the activity management application provides invaluable formative assessment information, which has been demonstrated to greatly improve student learning ([2], [7], [38]). For all of these reasons, SMILE may provide a particularly effective means of promoting student-created questions in an engaging way.

The remainder of the paper is organized as follows. SMILE is described in more detail in section 2, including summaries of how the program functions from the student's and the teacher's perspective. Section 3 describes a pilot study conducted in an elementary classroom along with its preliminary results. Finally, section 4 outlines brief conclusions and directions for future research.

2 Stanford Mobile Inquiry-based Learning Environment (SMILE)

2.1 SMILE network structure

SMILE consists of two elements: a mobile-based application for the students called Junction Quiz and an activity management application for the teacher called the Junction Quiz controller. As is depicted in Fig. 1, all components of this system are connected to an ad-hoc network, which introduces a temporary local connection supported by a wireless network router. This type of network is rapidly deployable, self-configuring, and does not require any existing network infrastructure. Open-source Apache software was used for the application server.

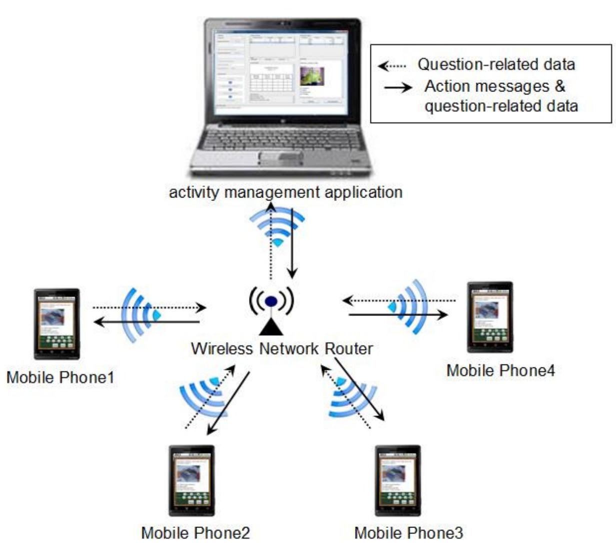


Fig. 1. Schematic of SMILE's ad-hoc network

2.2 Student's mobile-based application

The student's mobile-based application is called Junction Quiz. It was developed on Android 2.2 and can readily be installed on any wireless Android device. The main functionalities of the application are shown in Fig. 2 with seven screenshots (Fig. 2a-2g), which will be described in the section that follows.

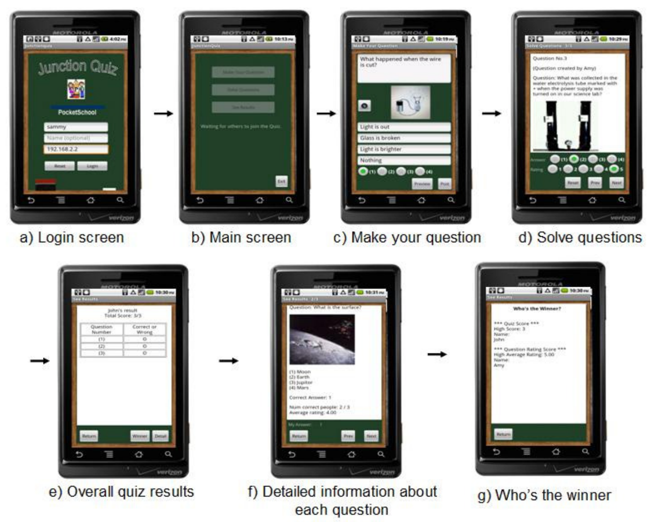


Fig. 2. Actions of mobile-based application

Upon opening the Junction Quiz application, students are first required to log in by entering their name at the login screen (Fig. 2a). They are then redirected to the main screen (Fig. 2b) which displays three grayed-out buttons, the first of which is "Make your Question". The button is grayed out until all students have logged in and the teacher decides to activate it via the activity management application. Once activated, the students proceed to create their questions. On the question creation screen (Fig. 2c), students write a question with four possible solutions. They have the option of including a new photograph or one already saved on the mobile phone. When all students finish writing their questions and their newly-created questions are sent to the server, the teacher activates the "Solve Questions" button. Students then solve questions and rate each on a five-point scale from 1 (poor) to 5 (excellent) based on some predetermined criteria (Fig. 2d). Again, once all students have completed this and the teacher decides, the "See Results" button on the main screen can be activated. At this point, students can view a summary of their results and see which questions they answered correctly or incorrectly (Fig. 2e). They can also view detailed information about individual questions including how many students answered each correctly and average ratings (Fig. 2f). Finally, students can view which student answered the most questions correctly and whose question received the highest ratings (Fig. 2g).

2.3 Teacher's activity management application

The activity management application for teachers, called the Junction Quiz Controller, serves to manage and save data from the activity via an ad-hoc network. It is a Graphic User Interface-based application that can be launched on and operated from any desktop, laptop, or netbook on which Java is installed. The application basically allows the teacher to control and monitor all of the students' activity in real time. It also allows the teacher to save the data from a given activity for later access. Fig. 3 is a screenshot of the activity management application. Important functionalities are highlighted with circular indicators labeled A-G on Fig. 3, each of which will be explained in the section that follows.

The Activity Flow window (indicator A) allows the teacher to activate the various stages of the activity such that students can only progress if the functionality has been enabled. The Student Status window (indicator B) displays the current status of each student on the present activity (i.e. who has joined the activity, who has submitted questions, and who has submitted answers). The Scoreboard window (indicator C) displays individual student's responses and ratings for each of the questions. The Question Status window (indicator D) displays which student created each question, what the average rating was for that question, and the percentage of the students who answered it correctly. The Question window (indicator E) displays the question itself and its predetermined correct answer. The Top Scorers window (indicator F) displays which student achieved the highest score and which question received the highest ratings. Finally, with the Save Questions button (indicator G), the teacher can save the data from a given exercise to the server, which can be accessed at a later date.

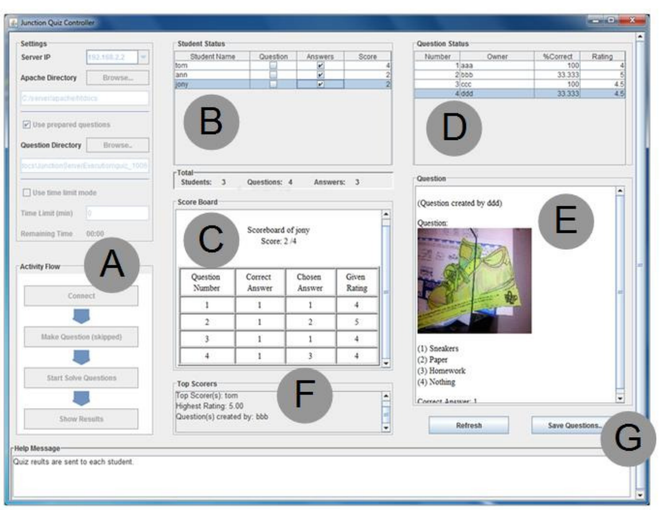


Fig. 3. Activity management application

3 Pilot study

3.1 Purpose

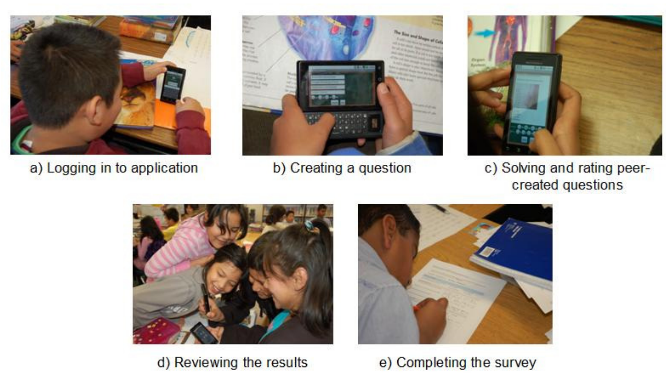
The purpose of this pilot study was to investigate students' reception to and perception of SMILE as a means of promoting student-created questions in an elementary classroom setting.

3.2 Participants

The participants in this study were 32 fourth- and fifth-graders (16 female, 50%) at a public school in California. The class was a hybrid composed of 14 high-achieving fourth-grade students and 18 fifth-grade students.

3.3 Method

Each student received one Motorola Android smart phone preloaded with the Junction Quiz application regardless of whether or not he or she already owned a phone. Students then listened to a 15-minute explanation of the application's basic functionality as well as what they were expected to do during the session. Fig. 4 includes photographs of the students completing each of the required activities during the pilot study. Students logged in to the mobile application (Fig. 4a), created a question (Fig. 4b), solved and rated peers' questions (Fig. 4c), reviewed the results (Fig. 4d), and completed a survey (Fig. 4e). The activity was contemporaneously managed using the teacher's activity management application. Due to technical and other difficulties six students were not able to submit their questions to the server in time, leaving 26 questions (84%) submitted in total.



a) Logging in to application
b) Creating a question
c) Solving and rating peer-created questions
d) Reviewing the results
e) Completing the survey

Fig. 4. Activity supported by SMILE

4 Survey & results

This section describes the results of the survey administered to the students following the activity. The survey included one closed-ended Likert-type question on overall satisfaction and five open-ended questions related to general student perceptions of the activity. Two boys did not complete the survey, yielding a 94% overall response rate.

4.1 Overall student satisfaction with SMILE

In order to evaluate overall student satisfaction, students were asked to indicate how satisfied they were with the activity on a four-point Likert scale from 1 (Very satisfied) to 4 (Not satisfied). As shown in Fig. 5 below, 87% of respondents were very satisfied with SMILE and 0% indicated that they were not satisfied.

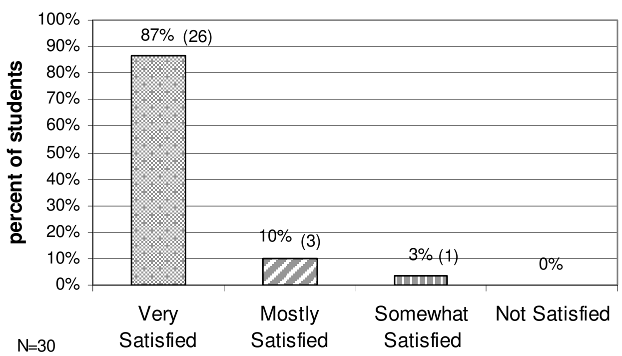


Fig. 5. Student Satisfaction with SMILE

4.2 Student perceptions of SMILE

In order to examine the students' perceptions in depth, they were asked the following open-ended questions:

(Q1) What do you like about this activity?

(Q2) What do you not like about this activity?

(Q3) How does this activity help you learn?

(Q4) What features would you like to add to this activity?

(Q5) What was the best thing about this activity?

Students' responses to each of these questions are discussed in the sections that follow.

Q1. What students liked about SMILE

Students' responses to Q1 relating to what they liked about the activity were coded and then categorized into four response types, which are displayed in Fig. 6. A majority of the students (73%) indicated that what they liked most about the activity was that it enabled them to create, share, and answer questions. For example, one student responded, "What I like most about this activity is that I could make up my own question and other kids could read my questions and answer them." Another said, "What I like about this activity is everything. It made learning notably fun!"

Q2. What students did not like about SMILE

Students' responses to Q2 relating to what they did not like about the activity were also coded and categorized into four response types, which are displayed in Fig. 7. The majority of students (60%) indicated that there was nothing that they did not like about the activity. For example, one student responded, "I like everything. Actually, I wouldn't change a thing or two." The remaining responses were informative and diverse in content. Two students indicated that they did not like having to wait for their peers to complete the tasks. Two others indicated that they disliked having to rate their peers' questions. Other responses were varied and included issues such as technical difficulties, low question quality, and question difficulty.

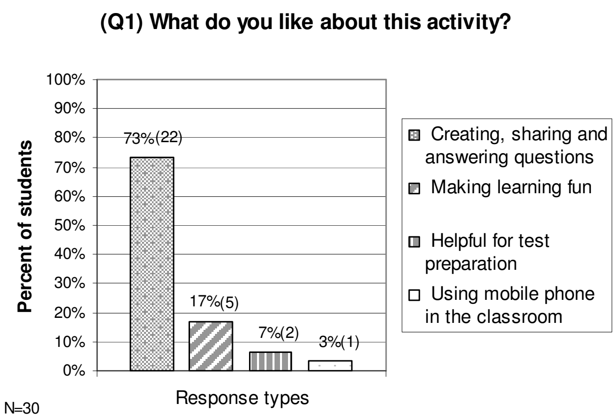


Fig. 6. What students liked about SMILE

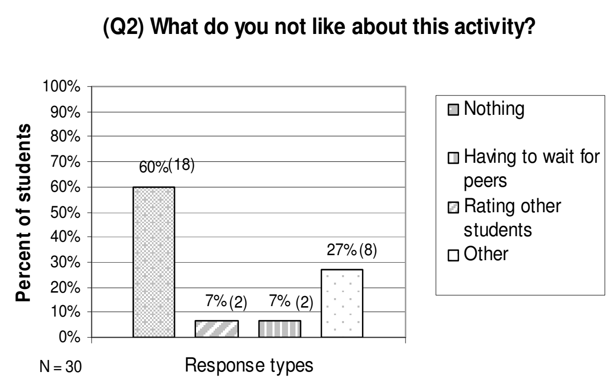


Fig. 7. What students did not like about SMILE

Q3. What students thought they could learn from SMILE

Students' responses to Q3 relating to how they thought SMILE helped them learn were coded and categorized into four response types as well, which are displayed in Fig. 8. The majority of students (60%) indicated that seeing their peers' questions helped them learn class materials. For example, one student said, "This activity could help me learn by answering the questions that I might not know and it could help me do right better" and another said, "It helps me learn a lot of new stuffs like questions I have never seen before." One student notably mentioned that this activity was valuable because it helped him learn to generate better questions. He said, "This activity helps me learn how to make good questions. It will make me know about good questions because I see other questions."

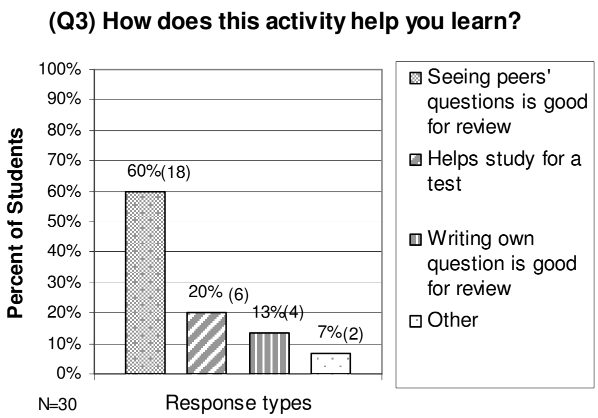


Fig. 8. What students thought they could learn with SMILE

Q5. What students thought was best about SMILE

Finally, students were asked what they considered to be the best part of the activity. Once again, students' responses were coded and categorized into different response types, which are displayed in Fig. 9. Similar to the results of Q1, the vast majority of students (87%) reported that what they liked best was creating their own questions and sharing them with peers. For example, one student wrote, "The best thing is that I can share the question I made with my friends and see my result." One student profoundly remarked that the best thing about the application was its portability.

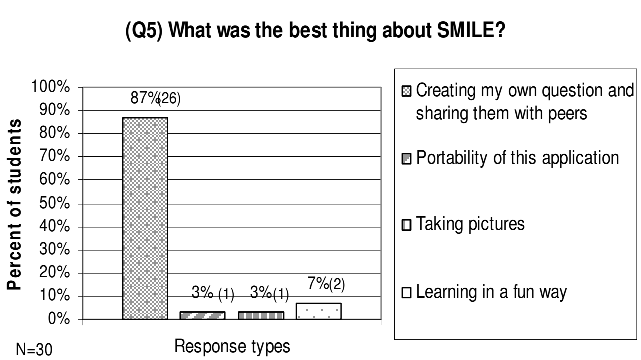


Fig. 9. What students liked best about SMILE

4.3 Student-created questions

Twenty-six students created questions for the activity, which varied in terms of sophistication. All questions were categorized into three question types drawn from Bloom's taxonomy [3]. The three question types that emerged were Knowledge-type (recall of data or information), Comprehension-type (state a problem in one's own words), and Analysis-type (distinguish between facts and inferences). Fig. 10 shows student examples of each type of question and Fig. 11 displays the distribution of these types among the students. The data show that Knowledge-type were the most common, followed by Comprehension-type, with Analysis-type questions being the least common.

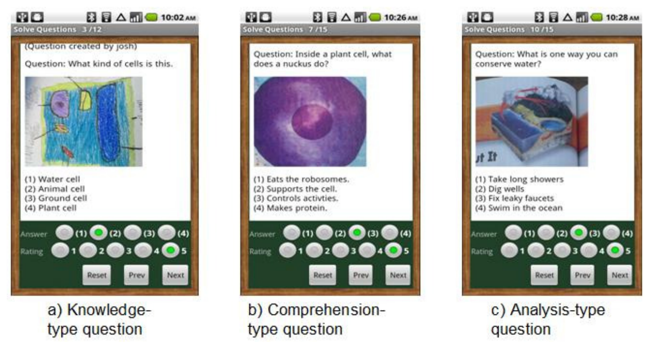


Fig. 10. Three types of questions created by students

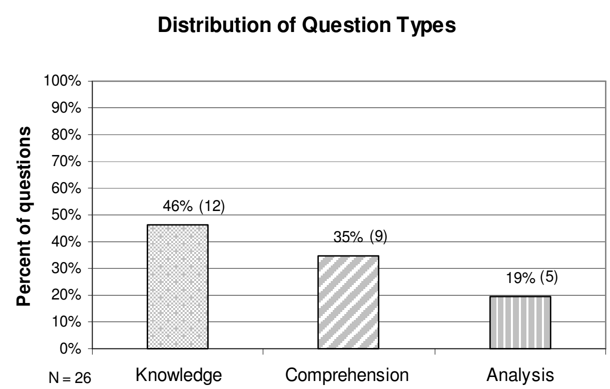


Fig. 11. Distribution of Question Types

5 Conclusion & future work

This paper describes the implementation of a mobile inquiry-based learning environment designed to promote student-created questions in an engaging fashion. Preliminary findings from this pilot study indicate that students were very satisfied with SMILE. Through open-ended questions, students specified that they most enjoyed the opportunity to create their own questions and share them with peers. Students also reported that they viewed SMILE as a valuable way to review class materials. Indeed, students created highly relevant questions for each other with a range of complexity spanning multiple levels of Bloom's Taxonomy [3].

In response to the positive feedback from the pilot study, we are planning to develop an expanded version of SMILE that allows students to independently access the server and engage with the application outside of the classroom. Additionally, although SMILE is presently a classroom-based learning environment, we hope to expand it to be a school-, district-, state-, and even nation-wide network of learners. With a larger community of students creating questions for each other, it will be possible to share student-created inquiries of the highest quality in the context of large-scale collaborative learning environments.

6 Acknowledgment

We would like to thank the teacher, Claudia Olaciregui, and her 32 students at Ellis elementary school for taking part in this pilot study. This work has been sponsored by the National Science Foundation grant (NSF#0832380, Programmable Open Mobile Internet) and the POMI research team at Stanford University.

7 References

[1] Barak, M. & Rafaeli, S., "On-line question-posing and peer-assessment as means for web-based knowledge sharing in learning," International Journal of Human-Computer Studies, Vol. 61, 84-103, July 2004.

[2] Black, P., & Wiliam, D., "Inside the Black Box: Raising Standards Through Classroom Assessment," Phi Delta Kappa, 1-13, Oct 1998.

[3] Bloom, B. S., Taxonomy of educational objectives, Handbook I: Cognitive domain. New York: David McKay, 1956.

[4] Cole, M., "Using Wiki technology to support student engagement: Lessons from the trenches," Computers & Education, Vol. 52, 141–146, 2008.

[5] Commeyras, M., "What can we learn from students' questions?" Theory into Practice, Vol. 34, Issue 2, 101-106, 1995.

[6] Corey, S. M., "The Teachers Out-Talk the Pupils," The School Review, Vol. 48, Issue 10, 745-752, 1940.

[7] Cross, K. P., "Classroom Research: Implementing the Scholarship of Teaching," American Journal of Pharmaceutical Education, Vol. 60, 402-407, 1996.

[8] Dale, E., "Children's Questions as a Source of Curriculum Material," Educational Research Bulletin, Vol. 16, Issue 3, 57-66, 1937.

[9] Dillon, J., "The remedial status of student questioning," Journal of Curriculum Studies, Vol. 20, 197-210, 1988.

[10] Dillon, J., The practice of questioning. London: Routledge, 1990.

[11] Dodds, R., & Mason, C. Y., "Cell Phones and PDA's Hit K-6," Education Digest, Vol. 70, Issue 8, 52-53, 2005.

[12] Dori, Y. J., & Herscovitz, O., "Question-Posing Capability as an Alternative Evaluation Method: Analysis of an Environmental Case Study," Journal of Research in Science Teaching, Vol. 36, Issue 4, 411–430, 1999.

[13] Gall, M. D., "The Use of Questions in Teaching," Review of Educational Research, Vol. 40, Issue 5, 707-721, 1970.

[14] Gilroy, M., "Invasion of the Classroom Cell Phones," The Hispanic Outlook in Higher Education, 56-60, 2004.

[15] Hunkins, F., Involving students in questioning. Boston: Allyn & Bacon, 1976.

[16] Kolb, L., Toys to Tools: Connecting Student Cell Phones to Education. Washington: International Society for Technology in Education, 2008.

[17] Marbach-Ad, G., & Sokolove, P. G., "Can Undergraduate Biology Students Learn to Ask Higher Level Questions? Journal of Research in Science Teaching, Vol. 37, Issue 8, 854-870, 2000.

[18] Mayer, R. E., "Multimedia Learning: Are We Asking the Right Questions?" Educational Psychologist, Vol. 32, Issue 1, 1-19, 1997.

[19] Mediamark Research & Intelligence, "Kids Intelligence," MRI report, 2010.

[20] Mosteller, F., "The "Muddiest Point in the Lecture" as a Feedback Device," On Teaching and Learning: The Journal of the Harvard-Danforth, Vol. 3, 10-21, 1989.

[21] Piaget, J., "The Role of Action in the Development of Thinking. In W. F. Overton, & J. M. Gallagher," Advances in Research and Theory. New York: Plenum Press, 1977.

[22] Reeve, J., & Deci, E. L., “Elements of the Competitive Situation that Affect Intrinsic Motivation,” Personality and Social Psychology Bulletin, 22(1), 24-33, 1999.

[23] Rothkopf, E. Z., “Learning from Written Instructive Materials: An Exploration of the Control of Inspection,” American Educational Research Journal, Vol. 3, Issue 4, 241-249, 1966.

[24] Shuler, C., “Pockets of Potential: Using Mobile Technologies to Promote Children’s Learning,” New York: The Joan Ganz Cooney Center at Sesame Workshop, 2009.

[25] Swan, K., van 't Hooft, M., Kratcoski, A., & Unger, D., “Uses and Effects of Mobile Computing Devices in K-8 Classrooms,” Journal of Research on Technology in Education, Vol. 38, Issue 1, 99-112, 2005.

[26] M. Birenbaum, & F. Dochy, Alternatives in Assessment of Achievements, Learning Processes and Prior Knowledge, Boston: Kluwer Academic Publishers, 93-129, 1996.

[27] Williams, E., “Student attitudes towards approaches to learning and assessment,” Assessment & Evaluation in Higher Education, Vol. 17, Issue 1, 45-58, 1992.

[28] Wilson, R. C., “Improving Faculty Teaching: Effective Use of Student Evaluations and Consultants,” The Journal of Higher Education, Vol. 57, Issue 2, 196-211, 1986.

[29] Denny, P., Luxton-Reilly, A., & Simon, B., “Quality of Student Contributed Questions Using PeerWise,” Proc. 11th Australasian Computing Education Conference, Vol. 95, 55-63, Jan 2009.

[30] Yu, F. Y., “Scaffolding student-generated questions: Design and development of a customizable online learning system,” Computers in Human Behavior, Vol. 25, Issue 5, 1129-1138, Sep 2009.

[31] Yu, F. Y., Liu, Y. H. & Chan, T. W., “A web-based learning system for question-posing and peer assessment,” Innovations in Education and Teaching International, Vol. 42, Issue 4, 337–348, 2005

[32] Yu, F. Y., “Promoting metacognitive strategy development through online question-generation instructional approach,” Proceeding of international conference on computers in education, 564-571, 2005.

[33] Mulye, R. & Westberg, K., “Mobile phone clickers in the classroom: experiences from a large introductory marketing class,” The World University's Forum 2009, Vol. 2, No. 5, 109-121. 2009.

[34] Cheung, S. L., “Using Mobile Phone Messaging as a Response Medium in Classroom Experiments,” The Journal of Economic Education, Vol. 39, No. 1, 51-67, 2008.

[35] R.E. Mayer, Cognitive theory of multimedia learning, The Cambridge handbook of multimedia learning, Cambridge University Press, 31–48, 2005.

[36] Lee, H. S., Liu, O. L. & Linn M. C., “Validating Measurement of Knowledge Integration in Science Using Multiple-Choice and Explanation Items,” Applied Measurement in Education, Vol. 24, Issue 2, 115-136, 2011.

[37] Strijbosa, J. W. & Sluijsmans, D., “Unravelling peer assessment: Methodological, functional, and conceptual developments,” Learning and Instruction, Vol. 20, Issue 4, 265-269. Aug. 2010.

[38] Brookhart, S., Moss, C. & Long, B., “Formative assessment that empowers,” Association for Supervision & Curriculum Development , Vol. 66, Issue 3, 52-57, 2008.

Using a Web 2.0 Approach for Embedded Microcontroller Systems

J. O. Hamblen[1] **and G. M. E. Van Bekkum**[1]
[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA

***Abstract** - This paper describes our experiences using a new approach for teaching an embedded systems design course and the associated laboratory. A cloud-based C/C++ compiler and file server are used for software development along with a low-cost 32-bit microcontroller board. Student resources include an eBook, web-based reference materials and assignments, an online user forum, and wiki pages with sample microcontroller application code. In laboratory assignments, breadboards are used to rapidly build prototype systems using the microcontroller, networking, and other I/O subsystems using small breakout boards with a wide variety of sensors, displays, and drivers. Software development is done in any web browser, all student files are stored on the web server, and downloading code to the microcontroller functions in the same way as a simple USB flash drive.*

Keywords: Embedded Systems, Design Project, Cloud Compiler, Microcontroller, Microprocessor

1 Introduction

Many schools offer an embedded systems design course. They can be found in electrical engineering, computer engineering, and in many systems oriented computer science undergraduate degree programs. For historical reasons, the course title can be somewhat different such as "design with microcontrollers" or "microprocessor based design" but they all have a lot in common.

These courses started out with low-cost 8-bit processors, but most of the development effort in industry has moved on to System on-a-Chip (SOC) 32-bit devices that contain a processor, volatile and non-volatile memory, and a wide assortment of I/O interfaces on a single chip. For software development in the embedded systems industry, the C/C++ family of languages is still the most widely used according to annual industry surveys [1]. Many new embedded devices now utilize networking even including those with microcontrollers.

This paper describes our experiences in developing such a course, updating it with new technology, placing an increased focus on networking, and utilizing an increased number of web-based resources. We will focus primarily on the technologies used in the associated student instructional laboratory for this course.

2 Embedded Systems Laboratory

This type of course requires a laboratory and is also an ideal place in the undergraduate curriculum to include design projects. Textbooks are always a bit problematic for such a course as once they include the details needed on a particular hardware setup for examples and laboratory assignments they quickly become out of date. We developed a textbook for the course that includes an overview of embedded systems design process with an emphasis on I/O systems, interfacing to external devices, and software development. This is where most of the development effort is focused for embedded products, now that a single chip microcontroller contains the processor, memory, and numerous I/O interfaces. The course textbook is distributed freely to students in an electronic format and is updated each semester. Students are required to own a notebook PC at our school and are allowed to use the notebook on tests as an eBook reader. Only a small percentage of students print or purchase a hardcopy of the textbook.

We use a microcontroller for the first half of the semester in the student laboratory assignments. The textbook covers the embedded development process, popular I/O interfaces, sensors, drivers, and networking. All materials for the laboratory assignments are provided on the web and can be updated each semester [2].

3 Laboratory Equipment and Tools

Selection of the hardware and software for a student laboratory is always a difficult decision. The desire to use the latest technology pushes course instructors to constantly update such a course. The cost of new equipment and software tools always makes the process more daunting. By adopting the some of the newer approaches being used in industry, students should be more productive and able to produce prototypes of complex embedded devices in less time.

For every desktop computer, there are around one hundred times as many processors found in embedded products worldwide. So job opportunities for students are vast in the embedded arena.

ARM processors are the most widely used processors in embedded devices. ARM does not make processor chips, but they license their processor design to over one hundred semiconductor manufacturers. They would be a natural choice for such a course. The Keil tools C/C++ compiler and emulator is one of the more popular development platforms for ARM processors in industry, but it is still somewhat expensive for schools. There are also some open source options for ARM C/C++ compilers.

We considered several options for the laboratory projects. One option we had considered initially was the new Cypress PSOC 5 ARM-based processor, but it had production delays and was not available in time for our course. We finally chose the mbed module seen in Fig. 1[3-5]. The small low-cost mbed module contains an NXP LPC1768 SOC processor. The new LCP1768 contains an ARM Cortex M3 processor, 64K RAM memory, a 512K Flash memory, a network controller, and a wide range of I/O interfaces such as USB, SPI, I2C, GPIO, ADC, DAC, RS232, and PWM. The price of the mbed module is significantly less than most textbooks, so it is even possible to consider the option of students purchasing their own mbed module. They are currently available for purchase from a number of web-based electronics distributors such as Digikey and Sparkfun.

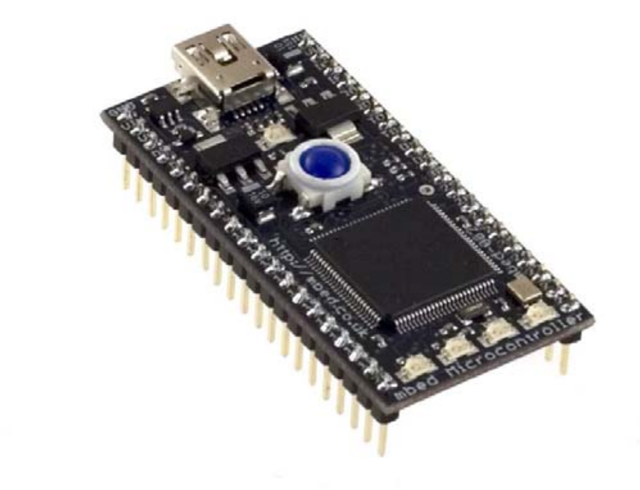


Fig. 1. The small low-cost mbed module plugs into a standard student breadboard.

The mbed module grew out of an internal research project at ARM to make the embedded development process easier both in industry and at schools [4, 5]. The module itself can plug into a standard student solderless breadboard. A USB cable connects it to a PC and for downloading code during software development it functions just like a USB flash drive. Power is provided by the USB cable. The USB interface can also function as a virtual com port allowing mbed programs to perform "printfs" or "scanfs" using any terminal application running on the PC.

For many years, schools have moved away from using solderless breadboards since many new ICs are only available in surface mount packages and not the older one tenth inch DIP style IC packages that plug directly into breadboards. Another problem with breadboarding was the large number of wires required to build up a realistic prototype of an embedded system. There is some pedagogical value in having students actually build up a circuit rather than using a large pre-assembled board. A breadboard also allows students to add their own custom hardware.

In the past couple of years, two factors have combined that make breadboards an interesting option to consider again for student laboratory work. Modern SOC processors already have sufficient internal memory and I/O interfaces on-chip. Most external I/O sub systems for embedded devices (i.e., sensors, displays, drivers, and networks) now use a serial interface that requires only a few wires. Due to a greatly increased level of hobbyist activity with the new generation of inexpensive single-chip microcontrollers, a large assortment of low-cost external I/O devices are commercially available pre-assembled on small printed circuit boards (i.e. breakout boards) that contain new surface mount ICs. They have pins that will plug directly into a standard student breadboard as seen in Fig.2.

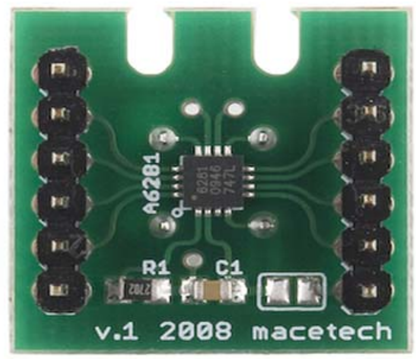


Fig. 2. A surface mount IC on a small breakout board that plugs into a student breadboard.

We compiled an extensive list of over one hundred commercial breakout boards with sensors, displays, drivers, and I/O connectors and posted it on the mbed wiki site [6]. A student breadboard project built with breakout boards is seen in Fig. 3.

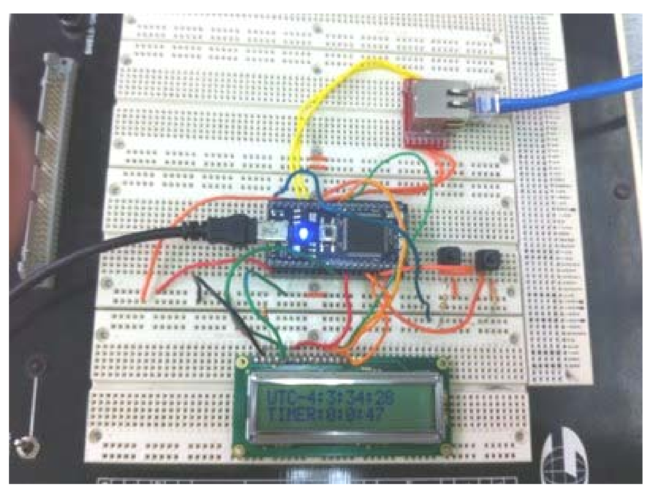


Fig. 3. An Internet Clock student project built using mbed and breakout boards on a breadboard. The time on the LCD is automatically synchronized with a NTP server via the Internet.

For those that want a pre-assembled board option there are also a number of commercial baseboards available for the mbed module [7]. The mbed module plugs into the baseboard. The baseboard typically provides an Ethernet network connector, microSD card slot, and a USB connector perhaps a sensor and small prototyping area as seen in Fig.4. On most baseboards, the prototyping area is limited. If students are going to design and add significant custom hardware in their laboratory projects, the breadboard may be a better choice.

Fig. 4. One of the commercial baseboards with network and I/O connectors designed for the mbed module.

4 Software Development

One of the more novel approaches of the mbed project was the decision to develop and support a cloud-based compiler for C/C++ software development. The compiler can run in any web browser as seen in Fig 5.

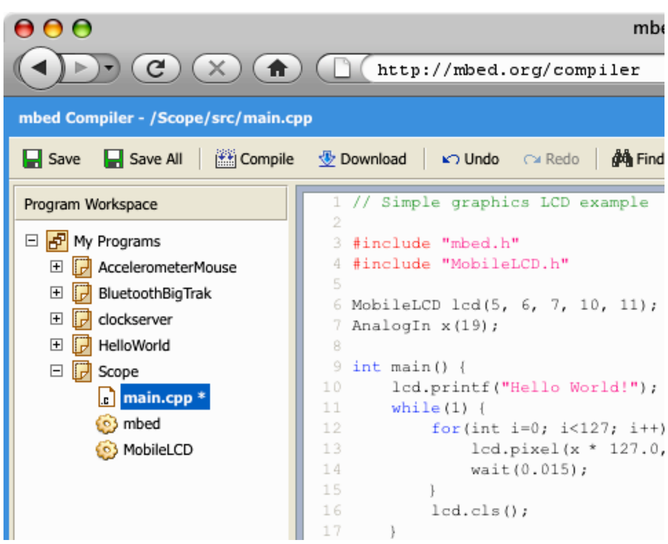


Fig. 5. The cloud compiler for mbed runs in any web browser. It is based on the Keil tools compiler.

Since the mbed module mounts just like a PC USB flash drive, it is easy to save new executable files to program the device. It runs the file with the newest date whenever you push the reset pushbutton on the mbed module. Another smaller ARM processor provides the USB flash drive interface and controls the main processor chip for debug. The firmware in this second interface processor cannot be modified by the user.

All students in the class can get a free password to setup an account on the cloud compiler with space for file storage [7]. Course instructors can request and obtain these passwords via email. Source files and documentation are saved on the mbed server. This means that there is no software to install and maintain, and development can move anywhere to any machine with a web browser. So students can easily work in the lab or at home on their projects. Most students can have a "hello world" application running on mbed in under five minutes.

In addition to the cloud compiler, the API support is also innovative. Network drivers, basic file system drivers, and easy to use APIs were developed for the NXP1768s on chip I/O features. These add higher level support for networking, files, PWM, SPI, I2C, Analog I/O, timers, delays, and RS232 serial ports using simple C++ object oriented library API calls.

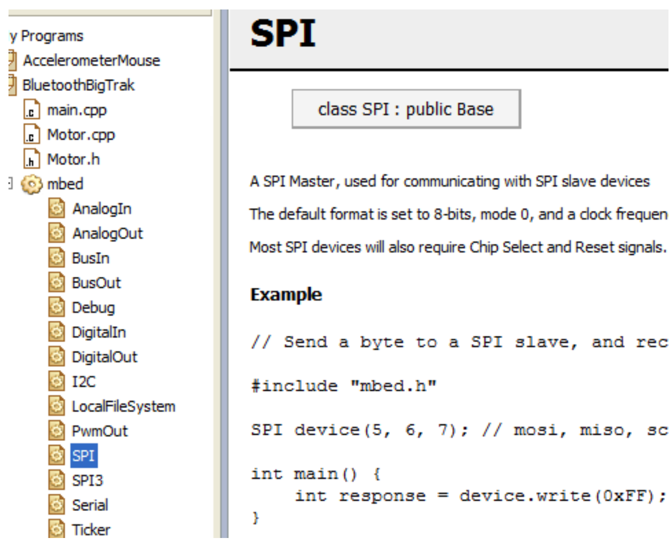


Fig. 6. The mbed API web-based documentation for an SPI interface and a code example.

Pin names can be used to specify the use of individual pins on the mbed module. Most pins can have four different programmable functions on the processor. The C++ object oriented APIs are able to automatically configure multifunction pins based solely on the pin names and the APIs used. Most students are able to hookup new I/O device hardware without ever checking the detailed data sheet for the device.

Along with the cloud compiler, the mbed website contains a number of helpful resources for students. Each student gets a notebook area where they can save documentation about a project. The handbook page contains an online API reference manual for the mbed I/O functions [9]. The documentation page for the SPI library API is seen in Fig. 6. There is an active user forum where students can post questions [10]. An extensive Wiki site contains documentation and code examples provided by users [6]. Code examples and projects can be easily imported from other users on the mbed web server with a few mouse clicks.

5 Laboratory Assignments and Projects

The topics for laboratory work will vary significantly at each school depending on the student's background and the goals of each individual course. Our class was oriented to embedded systems design and it contained a mixture of EE, CmpE, and a few CS undergraduates. Students had previously taken a digital logic design class, introduction to computer architecture, and C/C++ programming. Each laboratory assignment is two weeks for a total effort of around 6 hours. Students work in teams of two. Our institution provides Tsquare for web-based distribution of course materials and grades. Portions of Tsquare are based on the widely used Sakai Project. Tsquare was used to distribute mbed passwords, announcements, laboratory assignments, course materials, and grades.

In our course, we use mbed for two laboratory assignments followed by a short design project. We started out with a basic introductory lab where students used mbed for digital I/O, used PWM to dim an LED, added an I/O port expander, and used power management to reduce power levels by adapting C/C++ examples from the mbed wiki pages. For extra credit, they could add a watchdog timer or go back and use ARM assembly language instead of C/C++ for the basic digital I/O LED blink demo. Using assembly language made them appreciate the productivity gains using the C/C++ compiler and mbed's I/O API support. ARM assembly language development is possible using standard **.s* files in the cloud compiler [7].

The free evaluation version of the Keil tools compiler can also be used for I/O emulation and assembly language debugging offline. It is limited to 32K code size. The cloud-based compiler is based on the Keil tools compiler, so projects can be moved offline and back to the cloud compiler.

In the second laboratory experiment, students connected a number of different interfaces and devices by adapting several of the C/C++ cookbook Wiki code examples (i.e., RS-232, I2C, SPI, Analog in and out, USB, Ethernet, LCD text display, and a DC motor using PWM) and demoed each one working to the TA on a breadboard. The mbed's higher-level C/C++ I/O APIs really save some time here. In the class lectures at this time, we were talking about different I/O interfaces, so the lab was a good fit with the lectures.

After two introductory labs using mbed, we allowed students the freedom of a team-based design project where they could pick the idea. Instructor and TA approval and guidance regarding the scope of the project was required. Students post their project documentation using the mbed web site's notebook feature and they are encouraged to include photos and video clips. It is surprising the array of different ideas that were seen, and students also are more motivated when working on their own project idea.

6 Conclusions

There were very few software related issues to resolve. The cloud compiler with students keeping files on the server worked out better than many of our locally running tools and no support was required other than initially handing out student passwords and enabling network access for the mbed modules.

One concern we had initially was the support for debugging. Hardware breakpoints are not currently supported. To debug, the mbed module has four user LEDs, and "printfs" can print to a terminal application program running on the PC. Most of our problems occurred from students not wiring up all of the jumper wires correctly and not issues with debugging program code.

It is also possible to compile, set breakpoints, and emulate code and I/O offline using the Keil Tools compiler. Only a couple of students doing low-level hardware coding for their design project have needed to use this approach. The editing features available in the browser-based cloud compiler are a bit limited, but they seem to be rapidly improving with time.

Students need to pick a design project idea early in the term to allow time for any custom parts that might have to be ordered to arrive. It is also a good idea to remind students to check and only order parts that are in stock so that they arrive in time.

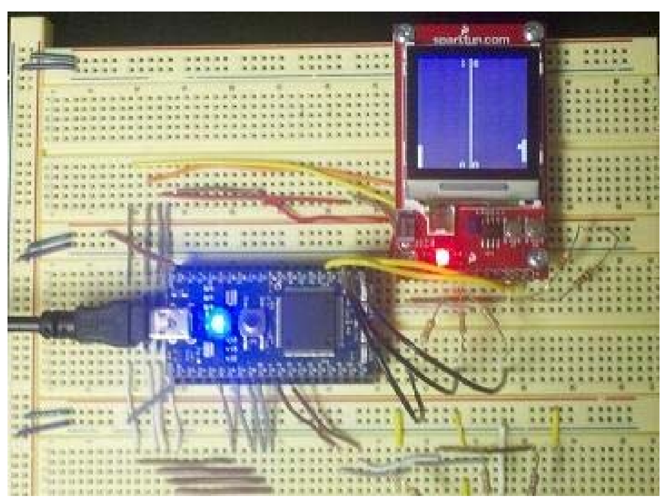


Fig 7. A student pong game project using a color LCD breakout board. The LCD is similar to those found in many cell phones.

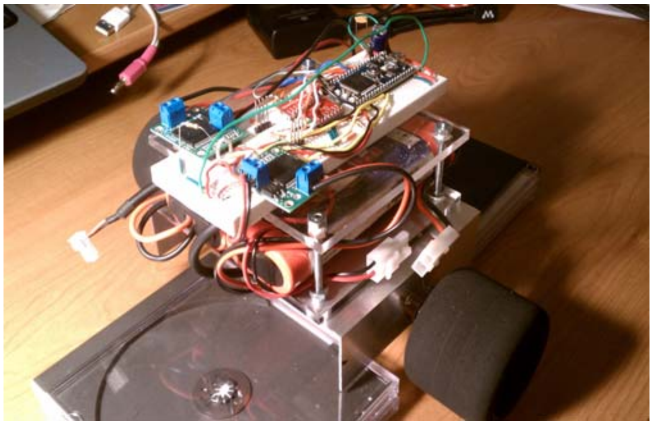


Fig 8. A two-wheel self-balancing robot project using DC motors with H-bridge breakout boards using PWM speed control, quadrature encoders, and a MEMs gyro and accelerometer breakout board with a PID control loop.

Fig. 9 An mbed-based internet radio built using Ethernet, USB, stereo audio jack, and MP3 audio decoder breakout boards.

Supporting all of the different design projects will require a larger assortment of sensor, drivers, and breakout boards than a more structured lab. The good news is that all of the parts unplug and can be reused for next term minus the few that get destroyed or lost. Surprisingly, so far with almost a hundred students no one has destroyed an mbed module. We did have a USB cable short out and lost a couple of the breakout boards.

Several examples of student design projects are seen in Figs. 7, 8, and 9. They include a pong game using a color LCD, a self-balancing robot, and an internet radio. All were built using the mbed module on a student breadboard using only commercially available breakout boards and jumper wires. Examples of many of the design projects from the class can be found in the mbed site's wiki pages under Student Projects [11].

A wide range of interesting devices were successfully prototyped for the design projects and approximately half of the projects used the internet. A number of students have also chosen to use mbed again in their team-based senior design project after taking the class.

7 References

[1] M. Barr, "Real men program in C", Embedded Systems Design, 2009 [Online]. Available: http://www.embedded-systems.com/design/218600142

[2] ECE 4180 Embedded Systems Design [Online] Available: http://www.ece.gatech.edu/~hamblen/4180

[3] Ashlee Vance, "You Too Can Join the Internet Of Things", New York Times, September 20, 2010. Available: http://bits.blogs.nytimes.com/2010/09/20/you-too-can-join-the-internet-of-things/

[4] S. Ford, Rapid Prototyping for Microcontrollers,[Online]. Available: http://mbed.org/media/press/mbed_whitepaper.pdf

[5] ARM University Program [Online]. Available: http://www.arm.com/support/university/

[6] J. Hamblen, "IC Sensor and Driver Breakout Boards" [Online]. Available: http://mbed.org/cookbook/IC-Sensor-and-Driver-Breakout-Boards

[7] Mbed Cookbook Wiki [Online]. Available: http://mbed.org/cookbook/Homepage

[8] Mbed Educational Program [Online]. Available: http://mbed.org/handbook/Education

[9] Mbed Handbook [Online]. Available: http://mbed.org/handbook/Homepage

[10] Mbed Forum [Online]. Available: http://mbed.org/forum/

[11] J. Hamblen, "Mbed Student Projects", 2011 [Online]. Available: http://mbed.org/cookbook/Student-Projects

Use of Web Multimedia Simulation in Biomedical Teaching

Jiří Kofránek, Pavol Privitzer, Marek Mateják, Stanislav Matoušek
Charles University in Prague, First Faculty of Medicine, Institute of Pathological Physiology, Laboratory of Biocybernetics, U nemocnice 5, 128 53 Praha 2, Czech Republic

Abstract - *The Atlas of Physiology and Pathophysiology designed as a multimedia-teaching tool, which helps to explain the function of individual physiological systems, causes and symptoms of their disorders in a visual way through the Internet is one of the projects in which we want to utilize new opportunities of multimedia and simulation models. Development of the Atlas requires cooperation of many professionals: Starting from experienced teachers whose design provides the foundation of quality educational applications, system analysts responsible for creating simulation models for educational simulation games in cooperation with professionals in their field, artists creating the visuals, and finally up to programmers who "knit" together the whole application to achieve its final form. For the inter-disciplinary collective creation to be successful, specific development tools with sufficient technical support must be used in each phase of creation; such tools allow for component-based creation of simulation models, creation of interactive multimedia and their final interconnection into a compact unit based on the given design. The Atlas of Physiology and Pathophysiology is a freely available application.*

Keywords: Education, Multimedia, Simulation, WEB

1 Introduction - Schola Ludus for the 21st Century

"Tell me, I'll forget, show me and I may remember; involve me and I'll understand" – this ancient Chinese wisdom is also confirmed by modern learning methods, sometimes called "learning-by-doing", where simulation plays are widely applied. Simulation plays make it possible to test the behavior of the simulated object without any risk – for example, try to land with a virtual airplane or, as is the case of medical simulators, treat a virtual patient or test the behavior of individual physiological subsystems.

The connection of the Internet and interactive multimedia environment with simulation models provides quite new pedagogical opportunities, particularly when it comes to explaining complex interconnected relationships, active exercising of practical skills, and verifying theoretic knowledge. The old credo of John Amos Comenius "Schola Ludus" – i.e. "school as a play" [4] pioneered by this European pedagogue as early as in the 17th century finds its application in the incorporation of multimedia educational games in training courses.

Similarly as the theoretic foundation of an air simulator is based on an airplane model, medical simulators are based on a sufficiently truthful model of physiological systems in the human body.

2 Complex integrative simulators for medical education

Models used as the theoretic foundation of medical trainers include ***mathematical models not only of individual physiological subsystems, but also their interconnections, thus forming a more complex unit***.

Coleman and Randal [5] created the **model "Human"** intended especially for educational purposes. The model (implemented in Fortran) allowed for simulating numerous pathological conditions (cardiac and renal failure, haemorrhagic shock etc.), as well as the effect of some therapeutic interventions (infusion therapy, effect of some drugs, blood transfusion, artificial pulmonary ventilation, dialysis etc.). Recently, Meyers et al. [19] made the original Coleman's model available on the web using Java implementation. Extensive training simulator ***Quantitative Circulatory Physiology (QCP)*** [1] is an extension of the Human model. The simulator proved to be useful in the teaching practice [22]. The recent simulator ***HumMod*** (formerly called ***Quantitative Human Physiology - QHP***) [9, 10, 11] with its more than 4000 variables apparently represents today the most extensive integrated model of physiological regulations. The model includes a menu branched abundantly, and it supports the simulation of numerous pathological conditions including the effect of any therapy. Unlike the previous simulator QCP whose mathematical background is hidden from the user in the source code of the simulator written in C++, the simulator ***HumMod*** has taken a different way. Its authors decided to separate the simulator implementation and description of the model equations in order to make the model structure clear for a wider scientific community. The model ***HumMod*** is distributed in its source form as open source (the model and the simulator are available to the public at the website http://hummod.org). Its structure is written in a special XML language and incorporates 3235 files located in 1367 directories. Thanks to this fact, the model equations and their relationships are comprehensible with difficulty, and many research teams dealing with the development of medical simulators therefore prefer to use older models of complex physiological regulations for their further expansions – for example, the classical models of Guyton of 1972 [8], and Ikeda's models of 1979 [12]. This is the path taken, for example, by the SAPHIR (System Approach for Physiological Integration of Renal, cardiac and respiratory control) project international research team as the source texts of QHP model seemed very poorly legible and difficult to understand to the project participants [25]. Similarly, Mangourova et al. [18] recently implemented an older Guyton's model of 1992 [2] in Simulink, rather than the most recent (poorly legible for them) version of the model QHP/Hummod of the team of Guyton's collaborators and students.

We were not discouraged and have established cooperation with the American authors. We have designed a special software tool ***QHPView*** [15] that creates a clear graphic representation of the mathematical relationships used, from thousands of files of source texts of the model. Besides others, this has also been helpful in discovering some errors in the model ***HumMod***. Together with the American authors, we are of the opinion that source texts of the models that are the foundation of medical simulators should be publicly available given that they are the result of theoretic study of physiological regulations – then it is easy to find out to what extent the model corresponds to the physiological reality. The structure of our model called "***HumMod-Golem edition***" is published at the project website (http://physiome.cz/Hummod) in its source form, together with the definitions of all variables and all equations. Unlike the American colleagues, our model is implemented in Modelica, which makes it possible to provide a very clear expression of the model structure.

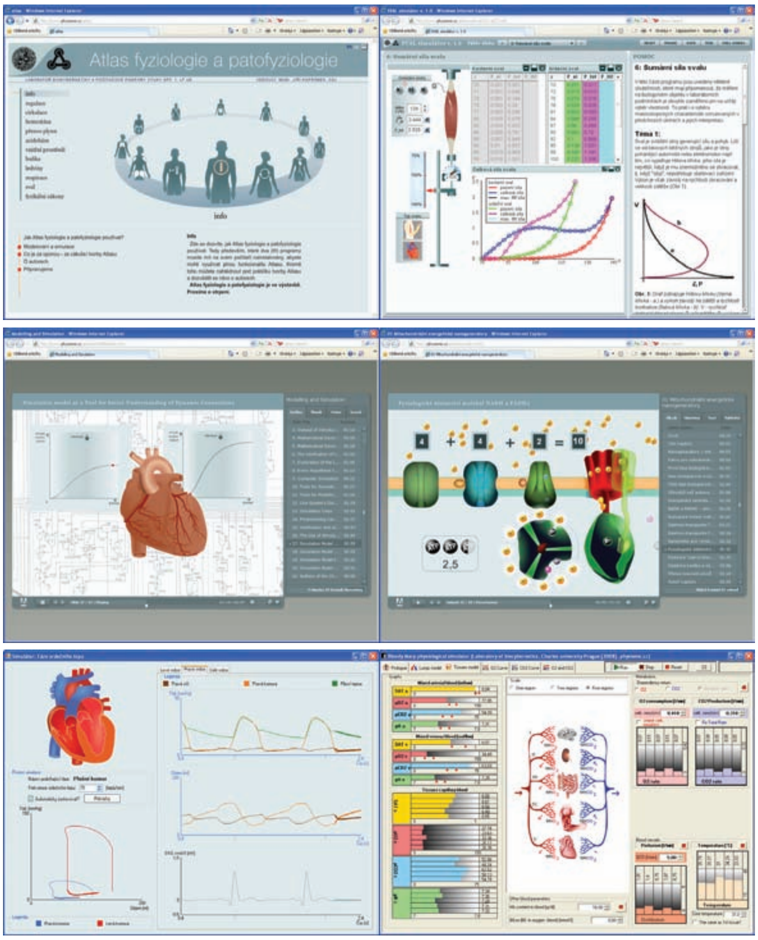


Figure 1: The Atlas of physiology and pathology combines interactive interpretations with sound, animations and simulation games. It has been created in Czech and (gradually) in an English version as well. It is freely available at: www.physiome.cz/atlas.

The model ***Hummod*** has been modified and expanded particularly in the field of blood gas transfer modeling and modelling of the homeostasis of the inner environment, especially of acid-base equilibrium – considering that disorders precisely of these subsystems occur frequently in acute medicine for which our simulator and educational simulation plays have been designed. Besides others, our modifications stemmed from our original complex model of physiological regulations, namely the core of the educational simulator ***Golem*** [13].

3 Atlas of Physiology and Pathophysiology - simulation games on the Web

However, experience in application of complex models (of the Golem or QCP type mentioned above) in teaching shows that large and complex models are connected with a disadvantage from the didactic point of view, namely their complex control. The large number of input variables as well as the broad scale of options of observing the input variables require rather thorough understanding of the very structure of the simulation model on part of the user, as well as knowledge of what processes should be observed in simulations of certain pathological conditions. In the opposite case, a complex sophisticated model seems to the user only as a "complicated and not very understandable technical play" (similarly as if the user should face a complex airbus simulator without a prior theoretical instruction).

Instruction models (and apparently not only complex ones with hundreds of variables) in themselves therefore are not enough for efficient use in teaching. They must be accompanied by explanation of their application – using interactive educational applications at best. The possibility of using all advantages of virtual reality to explain complex pathophysiological processes arises only upon ***establishing connection between explanation and interactive simulation***. In order to link the possibilities offered by interactive multimedia and simulation models in medical teaching, we have designed the concept of an Internet computer project, the ***Atlas of Physiology and Pathophysiology*** [13, 16], conceived as a multimedia instruction aid that should help to explain, in a visual way using the Internet and simulation models, the function of individual physiological subsystems, the causes and manifestations of their disorders – see http://physiome.cz/atlas. The Atlas thus combines **explanation** (using audio and animation) with **interactive simulation** play with physiological subsystems models, all available for free from the Internet.

Our Atlas is a part of the MEFANET network (MEdical FAculties NETwork), collecting electronic study textbooks and texts of medical universities in the Czech Republic and Slovakia Republic (http://www.mefanet.cz/index-en.php) The Atlas combines interactive lectures and chapters and simulation games with models of physiological systems (see Fig. 1). During the creation of the model user interface, used as a base for simulation games, the atlas looked more like an atlas of animated pictures from the regular, printed Color Atlas of Physiology [23] or the printed Color Atlas of Pathophysiology [24], rather than abstract regulation schemes used during biomedical classes. However, contrary to the printed illustrations, pictures creating the multimedia user interface in simulators are ***„alive“ and interactive – changes in parameters or variables will change the picture look as well.*** Thanks to interactive illustrations we may create simulation games which, better than regular still pictures or simple animations, explains the dynamical relations in physiological systems and help students to understand the causes or reasons that are involved in the development of pathogeneses in various diseases.

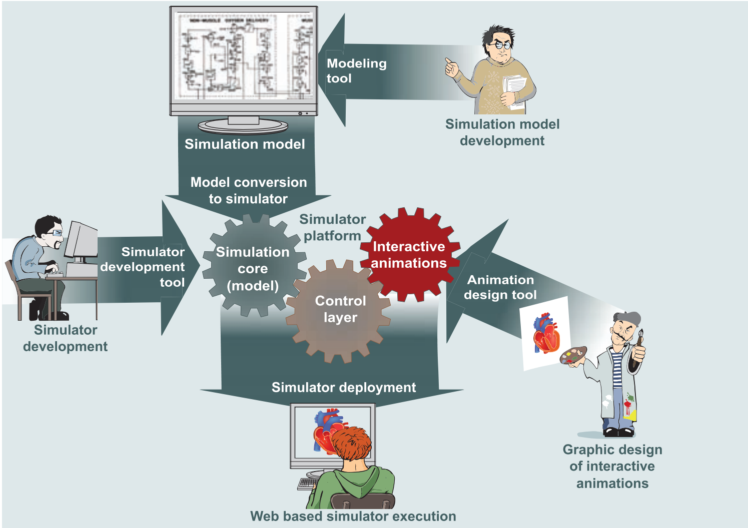


Fig. 2: Workflow of a web based simulator development and deployment. The simulation core is connected with interactive animation by a control layer. The model core is programmed manually or by means of automatic generation from a modelling tool (e.g. Matlab/Simulink or the Modelica-programming-language-based tool). Graphic components are created in Adobe Flash or Microsoft Expression Blend. Creating animations in Expression Blend offers the advantage of creating both the animations and the simulator in same .NET platform.

The Atlas uses the possibilities offered by the connection between interactive multimedia and simulation models, combines the simulations with tutorials, and simplifies the presentation layer while keeps the complexity of the model beneath. It is conceived as a multimedia instruction aid that should help to explain, in a visual way and using simulation models, the function of individual physiological subsystems and the causes and manifestation of their disorders. The Atlas thus combines an explanation of the physiological subsystems using audio and animation with interactive simulation (interactive for presentation layer, simulation for underlying model); it can be accessed at http://www.physiome.cz/atlas/index_en.html.

During the time of the Atlas development, it has been proposed new technologies allowing component-based creation of simulation models, creation of interactive multimedia, and their interconnection into a compact unit. The prerequisite for development of the Atlas was also the creation of a number of mathematical models of physiological systems and appropriate tools which enable to facilitate the design and sharing of the multimedia interactive educational simulators.

Atlas project is open-based, meaning its results are available for all those interested. During the course of its development, we welcome cooperation with all who would like to take part in its gradual building process.

4 From a mathematical model to a web-based multimedia simulator

Several key points should be considered when creating web-based multimedia enabled interactive simulators, such as those that are part of the Atlas of Physiology and Pathophysiology (see figure 2):

1. The underlying mathematical model needs to be formulated based on known physiological relationships. The model (i.e. set of equations that simulate the behaviour of the underlying object) is often implemented based on the verified models published in biomedical literature, but it can also represent original theoretical scientific work. In the past, simulation models were typically created in the same environment as the simulator itself (e.g. in the languages Fortran, C++ or Java). Today, special modelling tools can be used. Our team has been using the Matlab/Simulink (The Mathworks Inc.) environment on a long-term basis. We have created a Simulink library of formalized physiological relationships, named Physiology Blockset (available on http://www.physiome.cz/simchips). Recently, we have started implementing and creating mathematical models in an environment based on the language Modelica [15]. An essential innovation introduced by Modelica lies in its declarative and thus acausal definition of models. Individual parts of the model are described directly as equations and not as an algorithm to solve the equations. Modelica uses interconnected components in which equations are defined [6, 26].
2. The simulator itself is created based on the underlying mathematical model and the teaching goals; it is more programming rather than modeling work. In our case, the simulator generally has a three layer architecture known as MVC (model-view-controller) [3, 17]; the layers include the user interface with interactive animations, the control layer and the simulator core. The simulation core is obtained by converting a debugged model from the modeling tool (Simulink or modeling tool based on the Modelica language) to the

Table 1 - Atlas of Physiology and Pathophysiology: technologies used in the development of interactive simulator

Platform	Modeling environ-ment	Model conversion to simulator development tool	Simulator development tool	Animation development tool	Simulator deployment tool
Adobe Flash	Simulink	Manual	Action script based (Adobe Flash, Adobe Flash Builder)	Adobe Flash	Run in the Internet browser (with Flash Player plugin installed)
.NET	Simulink	Automatic	Microsoft Visual Studio	Adobe Flash	Installed localz from the Internet (by ClickOnce technology)
Silverlight	Modelica	Automatic	Microsoft Visual Studio	Microsoft Expression Blend, Animtester	Run in the Internet browser (with Silverlight plugin installed)

Table 2 - Atlas of Physiology and Pathophysiology: Comparison of various technologies used in the simulator development

Platform	Positives	Negatives
1. A simulator based on Adobe Flash platform	• No need for simulator installation, the simulator runs under various OS in the browser window (Flash Player plug-in needed). • Rich support for the visual aspects of the user interface.	• Manual conversion from Simulink to Action Script. • The simulation kernel is relatively slow.
2. A simulator based on .NET platform	• Automatic generation of the simulation kernel from Simulink. • Fast simulation kernel enables creation of computationally demanding simulators.	• Simulator runs under MS Windows OS only. • Need for simulator installation on the client computer.
3. A simulator based on Silverlight platform	• No need for simulator installation, the simulator runs under various OS in the browser window (Silverlight plug-in needed). • Automatic generation of the simulation kernel from a Modelica based environment. • Fast simulation kernel enables creation of computationally demanding simulators • Declarative model description (using model equations) in Modelica based environments. • Common platform of the user interface animations and the simulation kernel. • Animtester tool provides division between programming and the graphic design of the interactive animations.	• Silverlight plug-in is less widespread than FlashPlayer.

simulator development environment (ActionScript, Microsoft Visual Studio, etc.). The conversion can be done manually in the case of simpler models, but manual conversion of more complex models would be a tedious and error-prone job. We have created two software tools to convert the debugged models automatically (see Table 1). The control layer connects the simulation core with the interactive animations of the user interface and assures correct application logic. This layer is not needed in simpler simulators. The user interface is created in cooperation of the simulator programmer with a graphic designer and a teacher. It can evoke pictures from image-based medical textbooks.

3. In order for an educational simulator to look professional, a graphic designer should be the author of the user-interface animations. Our team has developed special tools that allow testing of the animation properties and subsequent connection of these interactive animations with the other model layers (e.g. Animtester made for Microsoft Expression Blend) - see figure 3. Hence, the artist is required to be proficient not only in standard technologies (e.g. Adobe Flash or Microsoft Expression Blend), but also in these testing tools. We have put substantial effort into the training of our artists in both areas.

4. An ideal means of educational material deployment has been the Internet. Uncomplicated user accessibility and relative ease of updates are among its advantages over portable electronic media. However, if a larger number of users get connected one-by-one to a more complicated simulation model then its placement on the server may bring problems with the server performance. In this situation, it is more effective to use the computing power of the client computers for running the model. We therefore used a technology, where the simulator is automatically downloaded, transparently installed and run securely in a Sandbox.

5 Web simulator creation technology

Three technology chains (and pertaining work-flows) have been used during the development of the Atlas. They are illustrated in Table 1, while Table 2 summarizes the advantages and disadvantages of each option.

Simulators based on a relatively simple mathematical model were implemented using the Flash Player platform, which allows them to be run directly in the browser window. Same platform was used for the explicatory lectures of the Atlas. The simulation kernel of the simulator was created in the Action Script language of the development tools Adobe Flash and Adobe Flex. Same tools were used for the design of the interactive animations.

More complex simulators were implemented using the platform .NET. We developed a software tool that allows automatic conversion of the mathematic model implemented in Matlab Simulink into language C#, and thus creating the simulator kernel. The created simulators can be installed on the client computer with one click using the technological solution ClickOnce [20]. The ClickOnce technology provides a way of running an application by clicking on an internet link. After confirming the security level, the application is automatically downloaded, transparently installed and run in a Sandbox.

The simulators implemented in the platform Silverlight are the result of our most recent work-flow and technological chain. These simulators can be computationally demanding and yet can be run in the browser window under various OS; the prerequisite is the Silverlight plugin in the Internet browser. The underlying mathematical model is first implemented in the modelling tool based on

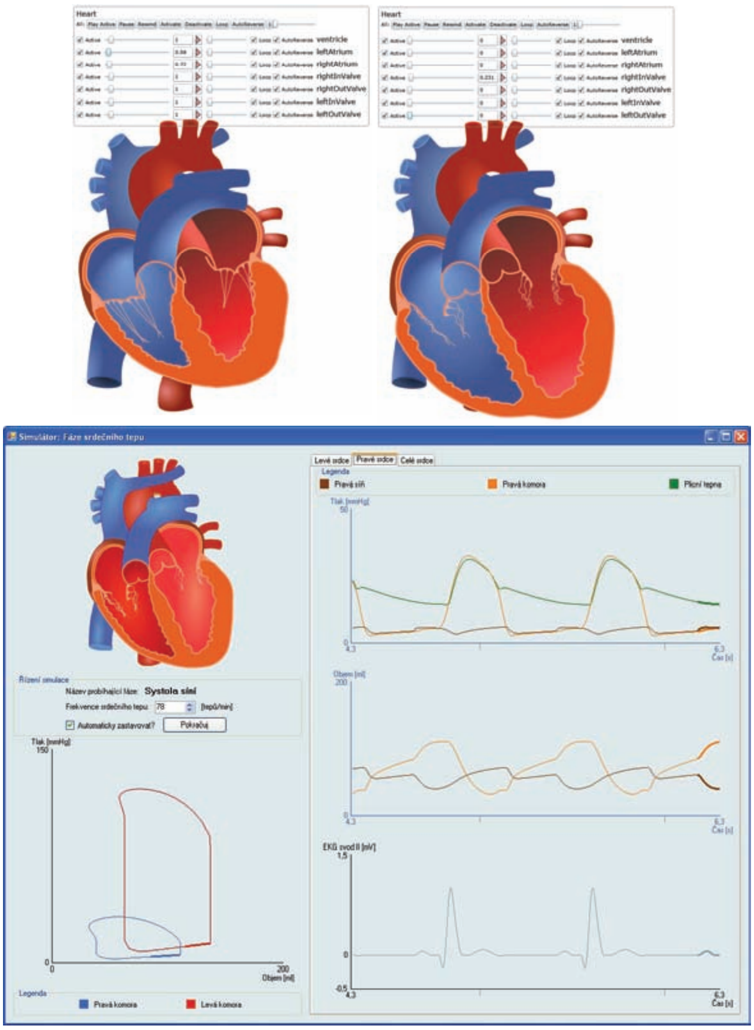


Fig. 3: Animation of a beating heart. Outputs from the model affect the phases of the heart pulse, opening and closing of cardiac valves, etc. Auxiliary Animtester control elements are above the animation and enable the graphic designer to set and tweak each sub-animation. The graphical designer is completely shielded from the programming process. In the final simulator, the "control ropes and levers" are pulled by the simulation model, programmed on the background.

the language Modelica. Typically, Modelica based tools can generate simulator kernels in the C++ language. However, simulators that contain parts of the code in C++ are not allowed to run in the browser window due to security requirements. As active participants on the Open Source Modelica Consortium [21], we have designed and implemented a code generator templating language that enables multi-targeting of the compiler output [7], and we have developed templates for C# code generation from Modelica models. This solution allows automatic conversion of a Modelica formulated acausal model into a C# formulated simulator kernel and we can produce pure .NET code able to run even under strict security requirements. Besides these improvements, the technology allows simulators of a more compact structure. The animations are created by graphic designers in Microsoft Expression Blend – a tool that communicates well with the rest of the platform. The already mentioned Animtester tool provides considerable support for cooperation of graphics and programmers. Its interface separates (and connects) the graphic design and simulator programming (see figure 3). The artist can create complex animations comfortably and the animations can be controlled easily. The programmer specifies the animation control by connecting it to relevant simulator modules.

The Atlas is composed of explicatory chapters and web-based simulators. The explicatory chapters of the Atlas are designed as audio lectures accompanied by interactive multimedia images (see figure 4a). Every animation is synchronized accurately with the explanatory text. Some simulators combine the model with the explicatory part. The simulator of mechanical properties of muscles is an example (see figure 4b).

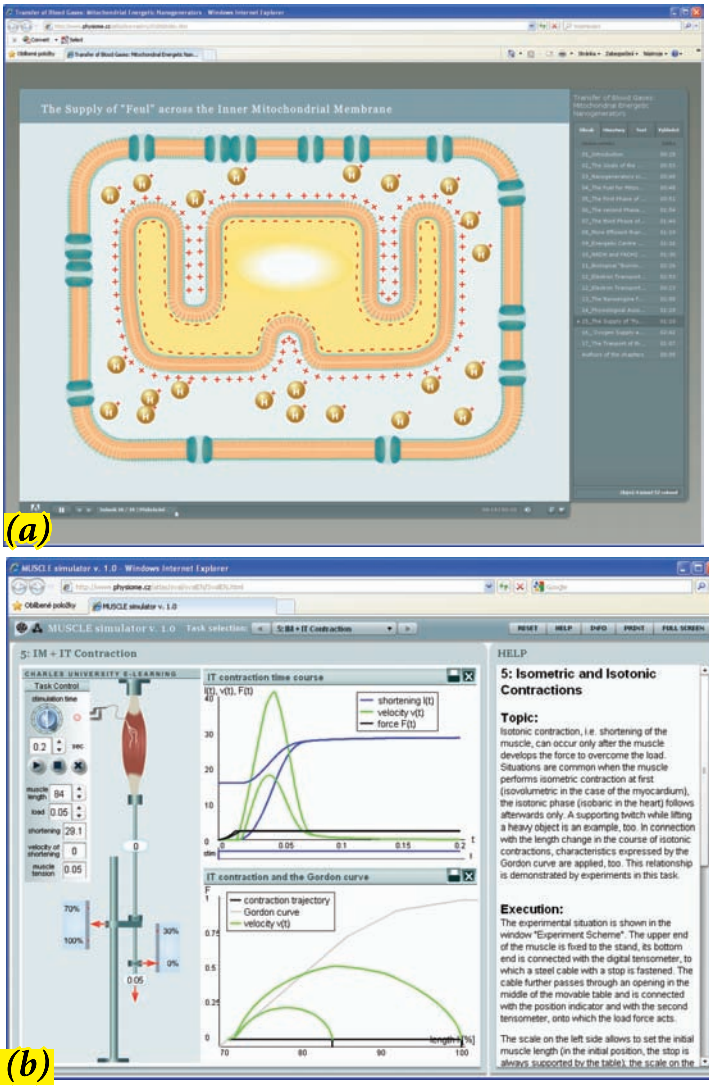


Fig. 4: (a) audiovisual interactive lecture in the explanatory part of the Atlas of Physiology and Pathophysiology. Every audio-explanation is accompanied by synchronized animated images. An explanation can be stopped at any moment in order to take a more detailed look at the accompanying animation. It can also be moved backward using the slide at the bottom of the player; (b) mechanical properties of skeletal muscle. Explicatory chapter including various simulation games. It is a Flash application. Accessible via http://www.physiome.cz/atlas/sval/svalEN/SvalEN.html.

Other simulators can be run separately and scenarios used in their control are planned as part of relevant explanatory chapters. The complex model of blood gases transport is an example; this model can be used as an instruction aid in explaining the physiology and pathophysiology of oxygen and carbon dioxide transport (e.g. to explain the consequences of ventilation-perfusion mismatch). This simulator can be downloaded from our Atlas using the following link: http://physiome.cz/atlas/sim/BloodyMary.

6 Conclusion – from enthusiasm to technology and multidisciplinary cooperation

The times of enthusiasts who created the first educational programs at the turn of the 80ies, excited about the new potential of personal computers, has long been gone. Today, the design of good-quality educational software capable of utilizing the potential offered by the development of information and communication technologies is not built on the diligence and enthusiasm of individuals. It is a demanding and complicated process of a creative team of specialists from various professions: Experienced teachers whose scenarios provide the foundation of a good-quality educational application; system analysts responsible, in cooperation with professionals of any given field, for the creation of simulation models for educational simulation plays; artists who design the external visual form of the simulator; and finally, information science specialists (programmers) who “stitch up“ the whole application to its final form.

For such interdisciplinary cooperation to be efficient, numerous developmental tools and methodologies are needed for every stage of development; such tools and methodologies make the work of individual team members easier and help them to overcome interdisciplinary barriers. Considerable efforts must be devoted to the process of creating and mastering the tools, but it pays in the end. The process of educational program design thus acquires ever more features of engineering design work.

7 References

[1] Abram, S.R., Hodnett, B.L., Summers, R.L., Coleman T.G. & Hester, R.L. (2007). Quantitative circulatory physiology. An integrative mathematical model of human mathematical model of human physiology for medical education, *Advanced Physiology Education,* 31: 202-210.

[2] cellML. (2010). *Description of Guyton 1992 Full cardiovascular circulation model*. [Online] http://models.cellml.org/exposure/cd-10322c000e6ff64441464f8773ed83/Guyton_Model_1-0.cellml/view

[3] Collins, D. (1995). *Designing object-oriented user interfaces*, Redwood City, CA: Benjamin Cummings, 1995.

[4] Comenius, J. A. (1656) *Schola ludus seu Encyclopaedea Viva*. Sarospartak, 1656.

[5] Coleman,T.G. & Randall, J.E. (1983). HUMAN. A comprehensive physiological model, The Physiologist, 26: 15-21.

[6] Fritzon, P. (2003). *Principles of object-oriented modeling and simulation with Modelica 2.1.* Wiley-IEE Press. ISBN 0-471-47163-1.

[7] Fritzson, P., Privitzer, P., Sjolund, M. & Pop, A. (2009). Towards a Text Generation Template Language for Modelica, *Proceedings 7th Modelica Conference*, Como, Italy, 193-207.

[8] Guyton, A. C., Coleman, T. G., & Grander, H. J. (1972). Circulation: Overall regulation. *Ann. Rev. Physiol.*, 41: 13-41.

[9] Hester, R. L., Coleman, T., & Summers, R. L. (2008). A multilevel open source model of human physiology. *The FASEB Journal*, 22: 756.

[10] Hester, R. K., Summers, R. L., Ilescu, R., Esters, J., & Coleman, T. (2009). Digital Human (DH): An integrative mathematical model of human physiology. *Proceedings of MODSIM World Conference NASA/CP-2010-216205*, 129-134. NASA, 2009

[11] Hester, R. L., Ilescu, R., Summers, R. L., & Coleman, T. (2010). Systems biology and integrative physiological modeling. *Journal of Physiology*, published ahead of print December 6, 2010, doi:10.1113/jphysiol.2010.201558, pp. 1-17.

[12] Ikeda, N., Marumo, F., & Shirsataka, M. (1979). A Model of overall regulation of body fluids. *Ann. Biomed. Eng.*, 7: 135-166.

[13] Kofránek, J., Anh Vu, L. D., Snášelová, H., Kerekeš, R., & Velan, T. (2001). GOLEM – Multimedia simulator for medical education. V L. Patel, R. Rogers, & R. Haux (Editor), *MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics.*, 1042-1046. London: IOS Press.

[14] Kofránek, J., Matoušek, S., Andrlík, M., Stodulka, P., Wünsch, Z., Privitzer, P., Hlaváček, J., & Vacek, O. (2007). Atlas of physiology - internet simulation playground. V B. Zupanic, R. Karba, & s. Blažič (Editor), *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation*, Vol. 2. Full Papers (CD ROM), MO-2-P7-5: 1-9. Ljubljana: University of Ljubljana

[15] Kofránek, J., Mateják, M., & Privitzer, P. (2010). Web simulator creation technology. *Mefanet Report*, 3: 32-97. Accessible on-line http://www.physiome.cz/references/mefanetreport3.pdf.

[16] Kofránek, J., Matoušek, S, Rusz, J., Stodulka, P., Privitzer, P., Matejak, M. & Tribula, M. (2011). The Atlas of physiology and pathophysiology: Web-based multimedia enabled interactive simulations, Comput. Methods Programs Biomed.(2011), doi:10.1016/j.cmpb.2010.12.007.

[17] Krasner, G. E. & Pope, S. T., (1968) A cookbook for using the model-view controller user interface paradigm in Smalltalk-80, *Journal of Object-Oriented Programming*, 1: 26-49.

[18] Mangourova, V., Ringwood, J., & Van Vliet, B. (2010). Graphical simulation environments for modelling and simulation of integrative physiology. *Computer Methods and Programs in Biome*dicine, Article in press, doi:10.1016/j.cmpb.2010.05.001, 10 pp.

[19] Meyers,R.D. & Doherty, C.L.. (2008). *Web-Human physiology teaching simulation (Physiology in health, disease and during therapy)*, Available online: http://placid.skidmore.edu/human/index.php).

[20] Microsoft, (2008). *ClickOnce Deployment for Windows Forms Applications.* [Online], Available: http://msdn.microsoft.com/en-us/library/wh45kb66.aspx.

[21] *Open Source Modelica Consortium* (2010), [Online], Available: http://www.openmodelica.org/index.php/home/consortium/

[22] Rodriguez-Barbero, A., & Lopez-Novoa, J. M. (2009). Teaching integrative physiology using the quantitative circulatory physiology model and case discussion method: evaluation of the learning experience. *Advances in Physiology Education*, 32: 304-311.

[23] Silbernagl, S., & Despopoulos, A. (2009). *Color Atlas of Physiology. 6th edition*, Georg Thieme Verlag, Stuttgart, 2009, ISBN 978-3-13-54006

[24] Silbernagl, S., & Lang, F. (2009). *Color Atlas of Pathophysiology 2nd edition*, Georg Thieme Verlag, Stuttgart, 2009. ISBN 978-3-13-116552-7

[25] Thomas, R. S., Baconnier, P., Fontecave, J., Francoise, J., Guillaud, F., Hannaert, P., Hernandez, A, LeRolle, V., Maziere, P., Tahi, F, & White, R. (2008). SAPHIR: a physiome core model of body fluid homeostasis and blood pressure regulation. *Philosophical Transactions of the Royal Society*, 366, pp. 3175-3197.

[26] Tiller M.,M. (2001). *Introduction to physical modeling with Modelica*. Kluver Academic Publishers, Boston, 2001. ISBN 978-9-7923-9367-4.

Arduino for Teaching Embedded Systems. Are Computer Scientists and Engineering Educators Missing the Boat?

Peter Jamieson
Miami University, Oxford, OH, 45056
Email: jamiespa@muohio.edu

Abstract— *In this work, we look at the Arduino as a design platform for a course on embedded systems and ask the question, is the Arduino platform suitable for teaching computer engineers and computer scientists an embedded system course with? To examine this question, we describe a project based learning embedded system course that we have taught and identify which topics are covered in it compared to the IEEE/ACM recommendations. The major contention lies in the idea that students can access and use an open source community that is focused on getting things working as opposed to strictly looking at low-level technical aspects of embedded systems. Additionally, the presence of open source and reusable designs makes it difficult to identify what a student is doing. In our experience, using the Arduino exposes students to sufficient complexity and challenges for an embedded system course.*

Keywords: Arduino, Embedded Systems, PBL

1. Introduction

In a recent article at Make online, titled, “Why the Arduino Won and Why It’s Here to Stay” [1], the author describes the world of microcontroller development kits and how the Arduino [2] has captured the hearts of many non-engineers. The question we pose in this paper is, should the Arduino and related open source projects be used as a platform to teach embedded systems?

To address this question we first try to establish which concepts/outcomes do we expect from a undergraduate level course in embedded systems. We then describe a course we have taught on embedded systems using the Arduino Uno, and discuss how the course does or does not satisfy these various topics. We find that the device as a platform, though not perfect, has many benefits that help students build devices that would not, likely, be possible with other control platforms. This is mainly due to the Arduino community, which consists of not only traditional engineers and scientists, but has a large contingency of artists and DIY hobbyists. The size of this community, the basic desire for users to get something working, and the open sharing of designs means students have access to a huge base of knowledge, that they can leverage to build their systems. We compare this with our experience in the previous year of the course where only FPGAs and PIC microcontrollers were available.

Even if using the Arduino kit allows students to avoid experiencing some of the low-level challenges of embedded system software and hardware design, the student still experiences a large majority of these concepts with the added benefit of building working and interesting designs. This point is debated from a philosophical standpoint in a number of areas. For example, should beginning programmers learn a language such as Java, which provides a rich library that quickly allows a student o create graphics, GUIs, and algorithms such as search and ordering, or should the beginning programmer be introduced to a language such a C and build up to these higher level concepts (even though C has these libraries, but they’re not part of a preset starting package and are easy for the novice to use). Similarly, should embedded system programmers and designers be provided a microprocessor that can only be programmed in assembly and has no existing framework or should an Arduino be provided that has a high-level compiler, a large community, and existing examples available on the Internet. Both choices have merit and we will discuss them in this paper.

The remainder of this paper is organized as follows. Section 2 attempts to identify what an embedded system course is and should cover. Section 3 introduces the Arduino device, the community and resources, and the advantages of using such a device. Section 4 describes the embedded system course at Miami University including how the Arduino can be used. Section 5 describes the projects created by the students and feedback collected from them. Finally, section 6 concludes the paper.

2. What is an Embedded System Course?

Grimheden and Törngren [3] asked this question in terms of defining embedded systems and asking how it should be taught. Their didactic approach asked a number of questions and their general conclusion was that embedded system courses are closer to functional and practical courses as opposed to disciplinary and formal. In Josefsson’s study on software engineers for industry [4] they identified the following skills of relevance:

- Business - budgeting and time management

- Software engineering/reengineering
- Teamwork
- Components reuse
- Human communication - written and spoken
- Testing and Validation
- Creating models

Project Based Learning (PBL) curricula (which is a version of Problem Based Learning) is becoming the norm for many engineering fields, business, and medicine [5], [6], [7] to help deal with the above identified topics in preparing students for the real-world. PBL pedagogy centers learning around the activity of the student. An approach to preparing students to become industrial designers is to include design projects throughout the curriculum, hence PBL curriculum. The accreditation agency, ABET, among other entities, influenced engineering programs into including a major capstone around 1995 to 1997 [7]. For computer engineering curriculum, lab only courses [8], [9] slowly evolved to include both labs and final projects. The senior capstone has been studied to help understand how to prepare students for this culminating experience [10], [11].

Embedded system courses fit well into PBL curriculum. The question still remains, what are the specific topics that should be covered in such a course. The IEEE/ACM model computer engineering curriculum [12] has an embedded system portion that consists of 11 units (7 core and 4 electives) with a total of 59 topics and 39 learning outcomes. The model does provide timelines, but a simple mathematical calculation leaves approximately 50 minutes per topic over a one-term 3 credit hour course to cover topics as complex as “DMA transfers” and “Memory system power consumption”. In other words, embedded systems is a very large subject matter for a single course.

The reality is a graduate degree in embedded systems as describe in ARTIST in 2003 [13] will, likely, cover all the topics in IEEE/ACM model over a number of courses, but the average computer engineer undergraduate will either need to extend their embedded system skills when in industry or they will never be involved in the field.

Other scholars have proposed course curriculum for embedded systems and embedded programming [14], [15], [16], [17], [18]. In all cases, the goal is to teach a common set of topics and to allow the student to interact with devices in the lab at both the software and hardware level.

An embedded system course deals with the design and analysis of the software and hardware for a dedicated application. In this review, we have identified that there is no common set of topics to cover for such a course, and there are a number of approaches to teach students a subset of these concepts at various depth of coverage.

3. Arduino and its Benefits

“Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It’s intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments” [2].

The basic system consists of a microcontroller with various peripheral interfaces that is programmed by an existing software platform. This may sound vague, but the Arduino can come in a number of form factors. For example, the Arduino Uno consists of an ATmega328 [19] microprocessor, a USB to serial chip, and an AC to DC power converter. The Uno can either be built by hand or can be bought premade from a seller such as sparkfun.com for approximately 35 USD. The Arduino software platform is written in Java and is based, mainly, on Processing [20] (a language developed mainly for artists). The IDE is installed on a machine and then can program the UNO over the USB. The base IDE includes a number of examples for blinking LEDs, making noises, etc. The Uno is only one type of Arduino kit and others exist such as the Nano (for compact use), the LilyPad (for wearable applications), and Fio (for wireless communication).

There is not much literature relative to Arduinos being used in computer science and computer engineering to our knowledge with the exception of [21]. Buechley *et. al.* [22] have described their experiences in building the LilyPad Arduino and using the device in workshops for K-12 based students. They noted how the device was successful in attracting females to participate in programming and hardware design. Also, Balogh [23] has described their Arduino based robot platform that was used in some lectures in robotic control and embedded systems in their curriculum. In their work, they noted the increase in Arduino searches on the Internet as a key factor in deciding to use the Arduino.

Taking a similar approach, Figure 1 shows the search trends for the terms arduino, x86, hc11, mips, and nios. What is remarkable is that Arduino searches are now roughly on par with x86 searches as of March 2011. This just illustrates the activity within the Arduino community.

The major benefits for using Arduino in an educational setting that we have identified are:

- Ease of setup - plug and play
- Many examples for controlling peripherals - preloaded in the IDE
- Many open source projects to look at
- Works on Windows, Linux, and Mac
- Low cost hardware - build or purchase prebuilt
- Low cost software - free
- Low maintenance cost - Destroyed microprocessors can be replaced for approximately 4 USD
- Students can prototype quickly
- Can be programmed in an a number of languages including C

On the other side of the argument, our two major concerns with using such a system are:

- What are the students learning and is this low-level enough?

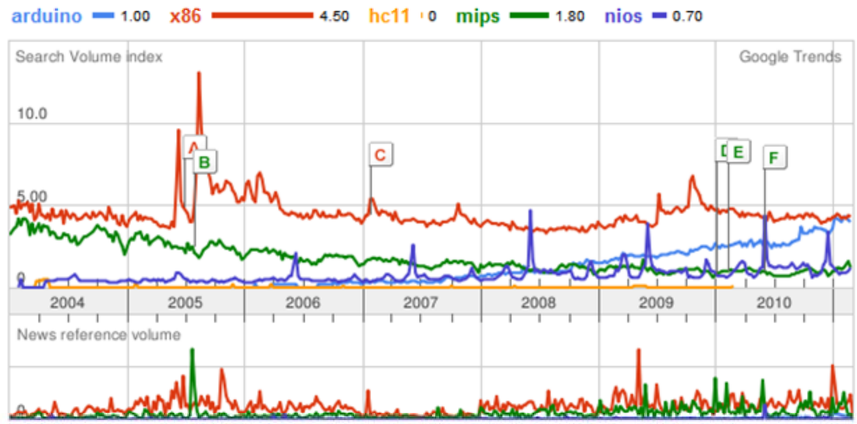


Fig. 1
SHOWS THE GOOGLE TRENDS FOR ARDUINO RELATIVE TO OTHER EMBEDDED TERMS

- How do we evaluate/assess external and internal open source contributions of the students design?

Open source software and hardware are relatively newer concepts in comparison to the timeline of computer science and computer engineering education. As to open source projects being included in our curriculum it is now the norm for undergraduates to be exposed to NIX operating systems via Linux among other systems. In 2002, Wolf *et. al.* [24] had one of the earlier debates on open source software and its relation to computer science education. Pedroni *et. al.* [25], recently, described their experiences with an open source programming project as part of one of their courses. They claimed that the students felt a greater satisfaction with the process, but had a tough time in the activity. In general, however, there is very little discussion on how to leverage open source projects and yet at the same time assess students in these projects.

In the next section, I will describe the current incarnation of our embedded system course and how it attempts to address embedded system content as well as our concerns.

4. Our Embedded System Course and how the Arduino is Used

Embedded Systems at Miami University is a course intended for third year Electrical and Computer engineers. The students have taken courses on programming, digital system design, computer architecture, and analog circuits. It is also possible the students will have taken courses in electronics, signals and systems, and advance programming including operating systems, but this is not guaranteed. The embedded course is a 4 credit hour course with 3 hours of lecture and 2 hours of lab time. The lab time is used as an open work time where the students spend their time on a number of projects.

The course is PBL focused and each student will prototype 3 embedded systems, will design 3 embedded systems (with the possibility of prototyping the system), and will present 3 times.

Table 1 shows these major activities in the course. Column 1 and 2 describe the activity type and what the goal is for the activity. Columns 3, 4, and 5 describe the group size, the time it should take the student in weeks, and if the activity is Arduino related. If the Arduino is “Possibly” used it means that the students have actually built prototyped their system, but this was not a course requirement.

For the three prototyping activities, the students have the choice of using the Arduino as there controlling device, and the students can choose what to build (where their proposal must be accepted by the instructor). The lab also has PIC microprocessors and DE2 FPGA prototyping boards (as part of Altera’s FPGA University Program [26]). Over 90% of the class used the Arduino for their midterm and final. The projects, however, were more varied as the students learned the limitation of a pin limited and computation limited device such as the Arduino. Various projects required either an FPGA, used the Xbox Kinect, and in some cases an interface with a PC.

The class activities are meant to help students design and understand embedded systems and various topics. The main teaching topics are understanding pin limits, cost and time for the alarm clock. In other words, how can a company profit from making something as simple as an alarm clock and what is needed beyond the basic electrical system. Many students designed their alarm clock and prototyped it on the Arduino even though this was not a requirement for the assignment. They stated that they just wanted to see the system work. The remote control activity is meant to teach a system with issues in power consumption, polling versus interrupt, and a communication protocol. Only one

Table 1
THE LIST OF ACTIVITIES FOR THE STUDENTS

Activity type	Activity Goal	Group Size	Weeks	Arduino
Midterm	Interface with another chip or device	1 to 2	4	Yes
Final	Build a simple embedded system	1 to 2	4	Yes
Project	Build an embedded system	2 to 4	8-12	Yes
Class Activity 1	Design an alarm clock	1 to 2	1	Possibly
Class Activity 2	Design a remote control	1 to 2	1	Possibly
Class Activity 3	Design a led display	1 to 2	1	Possibly
Presentation 1	Present a peripheral	1 to 2	2	Yes
Presentation 2	Present an embedded system	1-2	2	No
Presentation 3	Present your final	1	1	Yes

group pushed their design all the way to implementation on an Arduino. Finally, the last class activity focuses on bus communication protocols and real-time issues.

The presentation portion of the course has two goals. One to prepare students for communicating in the field, and two, to cooperatively use our numbers to quickly survey a number of chips and embedded systems so that the class can have a broad knowledge of the range of these devices.

The topics in embedded systems taken from the IEEE/ACM model computer engineering curriculum [12] that are not sufficiently covered in this approach are real-time operating systems, embedded multiprocessors, and networked embedded systems. The other topics are all covered in varying depth through the activities and classroom discussions.

To address our two concerns raised in the previous section, first, the depth to which the topics are covered varies from student to student and is dependent on the projects the students select. If a student did not choose a midterm chip that used a bus protocol such as I2C or SPI then the student's depth of coverage on the topic was significantly less compared to a student who did. However, in all the activities there is some topic that must be investigated at a deeper level. This is the nature of PBL and we believe the students get a far greater benefit from their own explorations as opposed to forced experiences.

In terms of how we evaluate projects that are connected to open source community, the first strict rule we enforce is that all external sources used (including fellow classmates) must be explicitly cited. As in all academic endeavors, it is possible that such a rule will not be followed. If a project uses a number of external sources and simply incorporates these sources together to form their project, we consider this a completely valid experience and submission, and therefore there is little incentive to copy without citation. The student might not benefit from developing specific pieces but system integration, the code reading and understanding, as well as integration skills are are useful and play a major part in real-world engineering.

5. Student Projects and Experiences

For the midterm, the students needed to find a chip of their choosing, get the device approved through an oral proposal, get samples of the chip, interface it with a controller, and add a wiki entry or webpage that could be used to help others use the chip. The idea behind this project is to make students aware of datasheets, sensors and actuators, and to use class resources to allow us to build a library of available chips and chip experts. The following chips were investigated by the students where Arduino has been highlighted if that was the control device used.

- SIS-2 IR Receiver/Decoder - **Arduino**
- Texas Instruments TLV5628: Octal 8-bit Digital to Analog Converter - **Arduino**
- ADXL-335 Analog Accelerometer - **Arduino**
- EDE1144 Keypad Encoder - **Arduino**
- FAN8082 Motor Driver - **Arduino**
- NA 556 Dual Precision Timer - **Arduino**
- AD8402 Digital Potentiometer - **Arduino**
- Texas Instruments TLC549cp Analog to Digital 8 bit Conversion Chip - **Arduino**
- MAX6969 LED drivers and piezzo buzzer - **Arduino**
- LM50 Single-Supply Centigrade Temperature Sensor - **Arduino**
- TMP01 Temperature Sensor combined with TLC549 Analog to Digital Converter - **Arduino**
- MC14021B NES controller - **Arduino**
- Servo Interfacing with the Arduino - **Arduino**
- HopeRF RMF12 (FSK Transceiver) - PIC
- XBox Kinect and the PS1080 SoC - PC and Kinect
- Texas Instruments TLC1543 (11 Channel - Analog to Digital Converter) - FPGA

In the previous version of this course where only PIC chips and DE2 FPGA prototyping boards were available, the students were not successful in interfacing with any chips except those that are on the FPGA board. We, therefore, modified the midterm requirements in 2010 to implement interrupt based audio interfacing with the DE2's Wolfson WM8731 Audio Codec chip and a NIOS II processor. Altera's university program [26] provides tutorials and sample code for this among other peripherals on the DE2. Unfortunately, the rapid change in Altera tools and the smaller university program committee make these tutorials only partially useful, and far worse in comparison with the Arduino community.

For the final, each student had to deliver a working demo of an embedded device of their creation that used at least 2 of the chips from the midterm or had some additional complexity. These projects included many simple robots that followed light, mapped the room, cleaned the floor, etc. Some novel projects included a time lapsed photography system, a Wii numchuck robotic arm, a 0-60 mph timer, and a speed golf swing analyzer. All of these projects were developed using the arduino as the main controller.

The major projects created for the class were varied as well, and the following projects were created where the Arduino impact is described:

- Automated parking garage - this is an Arduino project
- Remote control Nerfgun sentry - this is an Arduino project
- Funny walk robot - this is an Arduino project
- Guitar chord hero - this is an Arduino project
- Nerfgun auto tracking turret - this uses a PC to do the image processing and Arduino to control the turret
- Kinect body controlled remote cars
- LED dance floor with feedback - this is an FPGA based project with a front-end Arduino controller

As you can see by this list, Arduino is still a popular controller, but depending on the students desires (in particular image processing and high pin input/output systems) other control devices are used. This list of projects varies widely compared to the projects created in 2010, which included 3 alarm systems and 1 light control project. These 2010 projects only used either PIC (light control) or DE2 boards as the system controller, but otherwise were open projects of the students choosing. We believe that including the Arduino has allowed the students to create more interesting projects since they are not limited by some of the challenges of working with the DE2 or PIC. We should note that alarm systems were strictly banned from the 2011 choice of projects.

6. Conclusion

In this work, we have related our experiences in teaching embedded systems course while providing the students access to the Arduino platform and its open source community. We described the details of our course and showed how the Arduino can be used to expose the students to many of the topics normally included in an embedded system course. The majority of activities in the course are organized around the PBL pedagogy, and in all cases the students can choose to use any control platform, whether it be an FPGA, Arduino, or other microprocessor. The students have expressed high praise for the Arduino platform and we believe that their final projects compared to the previous years are better and more creative partially due to the availability of the Arduino kits. Access to the wikis generated by the students are available at: `http://www.users.muohio.edu/jamiespa/teaching.html`.

6.1 Discussion

We are pleased with the inclusion of the Arduino in our embedded system course. We, however, identify that the students are still missing two key components of their embedded system education. The first, as we have identified already, is coverage of real-time operating systems. To solve this situation, we, personally, would like to have an additional course on robotics in our department that would allow us to present this topic in relation to a realistic application.

The second missing topic in our current approach is software/hardware co-design. This element was included in the 2010 version of this course since FPGAs were used, but these concepts are now lacking as more and more of the students flock to the Arduino. Our belief is that this topic is a course in itself or can be incorporated into hardware acceleration or optimization courses. Again, the authors hope that such a course will be added to our departments curriculum in the future and are aware of a number of these courses being taught at the senior/graduate level.

References

[1] J. Provost, "Why the arduino won and why it's here to stay," Tech. Rep.

[2] Arduino, "Available at http://www.arduino.cc," 2010.

[3] M. Grimheden and M. Törngren, "What is embedded systems and how should it be taught?—results from a didactic analysis," *ACM Trans. Embed. Comput. Syst.*, vol. 4, pp. 633–651, August 2005. [Online]. Available: http://doi.acm.org/10.1145/1086519.1086528

[4] M. J. I. N. Institute, "Industriell Programvaruutveckling," Tech. Rep., 2003.

[5] C. L. Dym, A. M. Agogino, D. D. Frey, and L. J. Leifer, "Engineering design thinking, teaching, and learning," *Journal of Engineering Education*, vol. 94, pp. 103–120, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.1593

[6] J. Macias-Guarasa, J. Montero, R. San-Segundo, A. Araujo, and O. Nieto-Taladriz, "A project-based learning approach to design electronic systems curricula," *Education, IEEE Transactions on*, vol. 49, no. 3, pp. 389 –397, 2006. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1668283

[7] A. Dutson, R. H. Todd, S. P. Magleby, and C. D. Sorensen, "A review of literature on teaching engineering design . . ." *Journal of Engineering Education*, vol. 86, pp. 17–28, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.3949

[8] S. Areibi, "A first course in digital design using vhdl and programmable logic," in *Frontiers in Education Conference, 2001. 31st Annual*, vol. 1, 2001, pp. TIC –19–23. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.3048

[9] K. E. Newman, J. O. Hamblen, S. Member, T. S. Hall, and S. Member, "An introductory digital design course using a low cost autonomous robot," *IEEE Transactions on Education*, vol. 45, pp. 289–296, 2002.

[10] C. J. Lesko, Jr., “Building a framework for the senior capstone experience in an information computer technology program,” in *Proceedings of the 10th ACM conference on SIG-information technology education*, ser. SIGITE ’09, 2009, pp. 245–251. [Online]. Available: http://doi.acm.org/10.1145/1631728.1631804
[11] J. Goldberg, “Preparing students for capstone design [senior design],” *Engineering in Medicine and Biology Magazine, IEEE*, vol. 28, no. 6, pp. 98 –100, 2009. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5335729
[12] Joint Task Force on Computer Engineering Curricula, IEEE Computer Society, Association for Computing Machinery, “Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering,” Tech. Rep., 2004. [Online]. Available: http://www.computer.org/portal/c/document_library/get_file?p_l_id=2814020&folderId=3111026&name=DLFE-57612.pdf
[13] ARTIST, “Guidelines for a graduate curriculum on embedded software and systems,” Tech. Rep., 2003. [Online]. Available: http://www.artist-embedded.org/docs/Publications/Education.pdf
[14] J.-M. Vanhatupa, A. Salminen, and H.-M. Järvinen, “Organizing and evaluating course on embedded programming,” in *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling ’10, 2010, pp. 112–117. [Online]. Available: http://doi.acm.org/10.1145/1930464.1930484
[15] W. Wolf and J. Madsen, “Embedded systems education for the future,” *Proceedings of the IEEE*, vol. 88, no. 1, pp. 23 –30, Jan. 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=811598&tag=1
[16] H.-G. Gross and A. van Gemund, “The delft ms curriculum on embedded systems,” *SIGBED Rev.*, vol. 4, pp. 1–10, January 2007. [Online]. Available: http://doi.acm.org/10.1145/1217809.1217811
[17] D. Brylow and B. Ramamurthy, “Nexos: a next generation embedded systems laboratory,” *SIGBED Rev.*, vol. 6, pp. 7:1–7:10, January 2009. [Online]. Available: http://doi.acm.org/10.1145/1534480.1534487
[18] Y.-L. Huang and J.-S. Hu, “Hands-on oriented curriculum and laboratory development for embedded system design,” *SIGBED Rev.*, vol. 6, pp. 3:1–3:8, January 2009. [Online]. Available: http://doi.acm.org/10.1145/1534480.1534483
[19] *8-bit AVR Microcontroller*, ATMEL, 2011. [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf
[20] C. Reas, B. Fry, and J. Maeda, *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2007.
[21] J. Sarik and I. Kymissis, “Lab kits using the arduino prototyping platform,” in *Frontiers in Education Conference (FIE), 2010 IEEE*, 2010, pp. T3C–1 –T3C–5. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5673417
[22] L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett, “The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education,” in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI ’08, 2008, pp. 423–432. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357123
[23] R. Balogh, “Educational robotic platform based on arduino,” in *Proceedings of the 1st international conference on Robotics in Education, RiE2010*. FEI STU, Slovakia, 2010, pp. 119–122.
[24] M. J. Wolf, K. Bowyer, D. Gotterbarn, and K. Miller, “Open source software: intellectual challenges to the status quo,” in *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, ser. SIGCSE ’02, 2002, pp. 317–318. [Online]. Available: http://doi.acm.org/10.1145/563340.563464
[25] M. Pedroni, T. Bay, M. Oriol, and A. Pedroni, “Open source projects in programming courses,” *SIGCSE Bull.*, vol. 39, pp. 454–458, March 2007. [Online]. Available: http://doi.acm.org/10.1145/1227504.1227465
[26] AlteraU, “Altera University Program at http://www.altera.com/education/univ/unv-index.html,” 2010.

Electronic Collaborative Learning in Math-City

Kerri Stone, Irene Polycarpou, Julie Krause, and Cyndi Rader
Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, USA

Abstract— *Collaboration is a widely accepted and successful learning construct. Benefits of collaboration include improved academic achievement, increased motivation, and improved social skills. In recent years, electronic educational games have made their way into the elementary school classroom. Many of these games are constructed around individual or competition-based learning environments. Pedagogical research, however, shows that collaborative learning is a beneficial learning construct. As electronic games become popular in the classroom setting, the development of collaborative educational environments should not be overlooked. We present the design, development, and evaluation of Collaborative Math-City. Collaborative Math-City extends the Math-City electronic game and allows for a collaborative working environment among peers. The design of Collaborative Math-City incorporates both pedagogical research and educational software research.*

Keywords: Collaborative learning, educational electronic game, K-12 mathematics education, networked electronic game.

1. Introduction

Research on collaboration in educational software began in the mid-1990s. As early as 1996, researchers envisioned a networked collaborative electronic educational environment in which participants could collaborate over distances [1, 2]. A networked collaborative game can be used in a single classroom, across multiple classrooms in the same school, or even across school boundaries. Since the mid-1990s, several collaborative electronic educational games have been developed [3, 4, 5]. Additionally, researchers have developed guidelines for collaborative electronic interface development [1, 2, 6, 7, 8] and guidelines for successful collaborative electronic environments [1, 3, 9, 10, 11, 12].

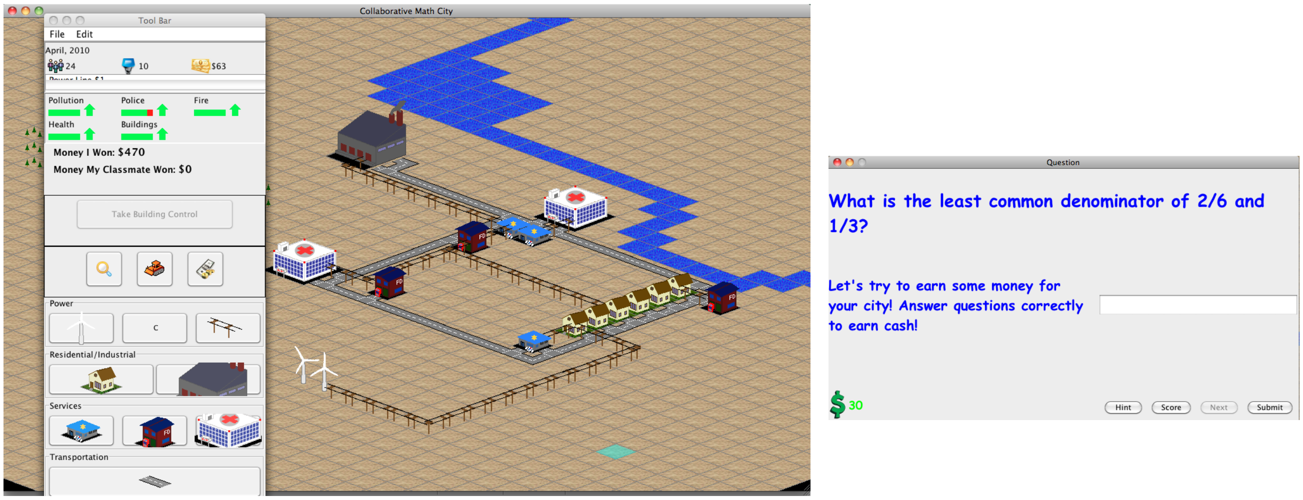
In this paper, we introduce Collaborative Math-City which extends the electronic educational game, Math-City [13], to allow for networked collaboration between students. Math-City is a simulation-based game in which students build and maintain a city, while practicing mathematics concepts and gaining exposure to renewable energy concepts. In the game, students answer grade-appropriate mathematical questions to earn money for building their city. Currently, Math-City includes mathematics questions related to fifth grade, but it is extendible to other grade levels as well. Students can use the money they have earned to purchase available items such as houses, police stations, fire departments, hospitals, roads, factories, coal plants, and wind farms to construct their city. The main goal is to develop a city that maximizes residents' happiness. Currently, the game includes five happiness indicators: Pollution, Police, Fire, Health, and Buildings (see game tool bar, Figure 1(a)). When the city becomes too polluted, safety issues arise (e.g, there are not enough fire departments), or there are too many large buildings (e.g., factories and coal plants) surrounding the homes, the residents of the city become unhappy. Students can improve the safety metrics by purchasing additional hospitals, fire departments, or police stations. They can also improve the buildings metric by placing large buildings farther away from the residences. Similarly, students can improve the pollution metric by removing factories or relying on renewable energy sources instead of coal plants.

Collaborative Math-City introduces the opportunity for students to work together toward the goal of creating and sustaining a city with happy residents. For the development of Collaborative Math-City, which uses a networked collaborative electronic environment, we leveraged research in human computer interaction (HCI), collaborative learning, and electronic educational tools. In what follows, we discuss the development of Collaborative Math-City as well as the results of two exploratory studies with K-12 students in a classroom-like environment.

2. Collaborative Learning

Pedagogical studies commonly research learning constructs and highlight differences between individualistic, competitive, and collaborative learning environments [1, 14, 15]. Students in an individualistic learning environment have no goal interdependence. The lack of goal interdependence means that "...individuals perceive achievement of their goal as independent of the other group members achievement" [1]. In a competitive learning environment, success is defined relative to classmates' performance; thus, competitive environments have negative goal interdependence, meaning that only a subset of students can achieve their goals [1, 15]. Students in a collaborative learning construct work together to achieve common goals; in a collaborative learning environment, "...there is positive interdependence among classmates, with each student achieving his/her own learning goals and outcomes only if other students in the group also achieve their goals" [1, p. 31]. Collaborative learning groups share the learning responsibility.

Collaborative learning environments, when compared to individualistic and competitive learning environments, gen-



(a) Example Collaborative Math-City city game board.

(b) Example Math-City math question.

Figure 1: Example Collaborative Math-City game board and math question.

erally result in stronger interpersonal relations and attitudes among students [15]. Furthermore, students working in collaborative environments exhibit improved self-esteem, have less difficulty following instructions, and waste less time waiting for teacher assistance [1]. These benefits of collaborative working environments can be summarized as fostering improved social skills and behavior.

The results of a study conducted by Hymel et al. [15] on the academic achievement and motivation realized in both collaborative and competitive learning constructs suggest that successful collaborative learning environments develop positive perceptions of competence in all participants. Moreover, the results of a study conducted by Zakaria et al. [16] suggest that collaborative learning methods not only improve students' mathematical achievement, but also improve their attitudes towards mathematics. In addition, after analyzing the effects of cooperative learning on student achievement in elementary school mathematics, Gilbert [17] concluded that while cooperative learning has no significant impact on academic success in grades one and two, it has a positive effect on academic success in grades three, four, and five. Since the current Math-City implementation is targeting fifth grade students, game collaboration is appropriate.

For easy collaboration, Collaborative Math-City is designed to support vocal communication and asynchronous participation, which are needed to better support group and individual tasks, respectively [3]. The individual task within Collaborative Math-City is an asynchronous task (i.e., answering math questions is an individual and asynchronous task); the group task within Collaborative Math-City, however, is defined as a synchronous task (i.e., building the city). Thus, Collaborative Math-City implements a clear turn-taking protocol to enable seamless synchronous interaction [5]. The following section details how we enabled collaboration within Math-City.

3. Design

Within Math-City, students have two primary tasks: answer math questions and build a city. Collaborative environments are most successful when the group task does not have a well-defined answer and can therefore be classified as an "ill-posed problem" [1, 9, 14]. The math questions in Math-City have well-defined answers and thus, are not "ill-posed problems." Successfully building the city, however, can be seen as an "ill-posed problem," and therefore should make for a strong collaborative or group task. As a result, we designed Collaborative Math-City such that students answer math questions on their own, but share the task of building the city. When a student selects to answer math questions to earn money for the team, s/he is presented with individual math problems (the collaborative partner does not see the same questions). Additionally, when one student in a collaborative pair is answering questions, the other student can elect to either answer questions or build the city. The Math-City city game board and an example math problem are displayed in Figures 1(a) and 1(b), respectively.

Collaborative Math-City currently supports collaboration between two students and students are restricted to sequentially building the city. In other words, students take turns contributing to the construction of the city. It is important to allow all participants in a collaborative learning environment to continuously contribute [2, 10]. Furthermore, according to the results of field studies conducted by Inkpen et al. [2] and Scott et al. [10], given the chance, collaborative pairs will do multiple activities at the same time. Collaborative Math-City supports continuous contribution and concurrent interaction by allowing students to answer math questions at any time during game play. Therefore, the turn-taking approach to building the city should not influence a team's ability to succeed or an individual's ability to contribute.

3.1 Turn-Taking

To avoid conflicts while updating the city landscape, only one player can add to the city at a time. This design decision was based on a literature review on effective turn-taking protocol for students at the elementary school level. Inkpen et al. [6] identify and explore the implementation of two turn-taking protocols: *give* and *take*.

Under the *give* turn-taking protocol, control is transferred from the active participant to his/her partner at will. Under the *take* turn-taking protocol, the inactive partner acquires control by taking it from the active partner. Research conducted by Inkpen et al. [6] shows that female students perform best in collaborative pairs while using the *give* protocol and that male students perform best while using the *take* protocol. Their findings also suggest that male student performance in collaborative pairs is actually hindered by the *give* protocol, whereas female performance is improved by the *give* protocol but is not drastically hindered by the *take* protocol. Due to the gender discrepancy in preferred turn-taking protocols, Collaborative Math-City implements both the *give* and *take* turn-taking protocols, with the *take* protocol as the default.

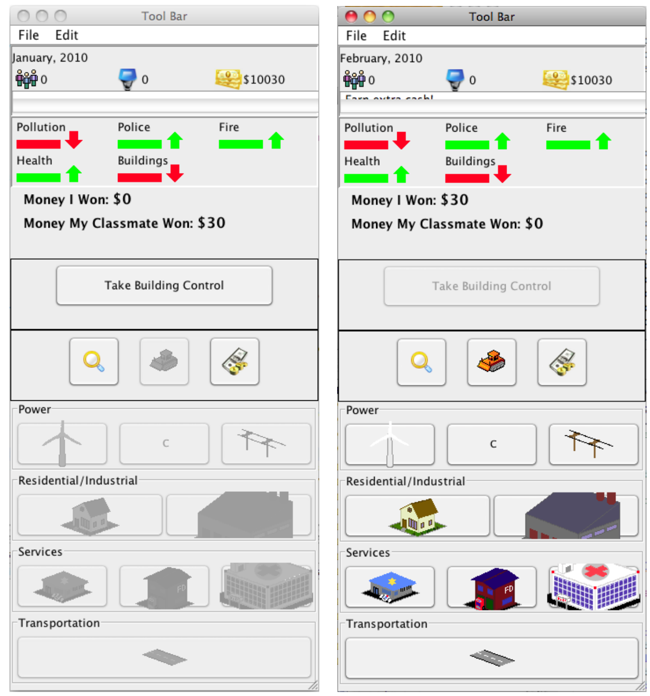
3.2 Rewards

Several researchers emphasize the need for collaborative environments to provide individual feedback, which can be referred to as individual accountability [15, 18, 19]. When working in a collaborative environment, it is important for students to know how much they have contributed to the team's success [1]. Research shows that individual accountability in collaborative environments increases a student's drive to continue on and keep learning [19]. Due to these findings on the need for individual feedback and accountability, Collaborative Math-City tracks the total amount of money each team member has contributed to the team's success (see Figure 2). Giving students a means to compare their performance to a social standard increases individual effort [18, 19].

In Collaborative Math-City, the social standard to which students can compare their own progress is the amount of money their teammate has earned.

3.3 Communication

Communication in collaborative electronic environments can be achieved in three ways: (a) text, (b) voice, and (c) video. Text-based communication is a useful tool that enhances and creates user awareness in collaborative environments [20]. Text communication, however, is most useful in synchronous collaborative environments [1]. Collaborative Math-City is asynchronous (i.e., one student can build the city while the other student answers math questions). Thus, we did not consider text-based communication solutions.



(a) Player 1 tool bar. (b) Player 2 tool bar.

Figure 2: Tool bars within Collaborative Math-City.

According to the research findings of Greenberg and Bohnet [21], voice communication works well in collaborative environments with four or fewer participants. Moreover, "...allowing people to see one another (i.e., video communication) does not add significantly to the process of collaboration." [1]. Due to these findings and the fact that this research does not envision Collaborative Math-City supporting more than four participants per group, we have implemented full-duplex vocal communication between participants within a group, and expect this vocal communication to supply an adequate collaborative environment for students. While playing Collaborative Math-City, students wear headphones to hear their partner talk, and communicate to their partner through a computer's built-in microphone or via an externally-connected microphone.

4. Implementation

The Collaborative Math-City implementation can be broken down into two main parts: initialization and synchronization. During the initialization phase, users have the ability to select to play Math-City in single-player or multiplayer mode (i.e., Collaborative Math-City). If Collaborative Math-City is selected, users are then guided through a process to setup a network connection. This network connection is used to synchronize the collaborative environment across computers. Synchronization can be broken down into two main parts that enable collaboration between participants:

(a) the *game state*, which encompasses logic to pass game information between participants' machines; and (b) the *vocal communication*, which encompasses logic needed to enable full-duplex vocal communication between participants' machines.

4.1 Game State

Collaborative Math-City synchronizes students' game boards using the transmission control protocol (TCP). TCP provides reliable and ordered delivery of data streams from a program on one computer to a program on another computer. TCP was selected for use in this part of the collaborative implementation because guaranteed delivery and synchronization order are important. The game state information is passed across the TCP socket connection in three formats: (a) *turn*, (b) *city*, and (c) *economy*.

To transfer control of building the city between players, a *turn* token is sent over the TCP socket connection. When it is not a student's *turn*, the student can answer math questions to contribute to the team's pool of money; however, students are not permitted to contribute to the city unless it is their *turn*. The game tool bar, which is displayed in Figure 2, includes a turn-taking button to pass control between participants' machines. In the *take* protocol, students cannot contribute to the city (building buttons are disabled) until they take control by pressing the "Take Building Control" button. In the *give* protocol, the button to pass control is labeled "Give Building Control," and the building buttons are enabled and disabled accordingly.

Constructed and demolished *city* elements are sent over the TCP connection to synchronize cities between players. When a player adds/removes elements (e.g., hospital, police station, fire department, house, road, etc.) to/from the *city*, the changes are passed across the TCP socket connection in the form of an associative array. This array is used to update collaborative partners' game boards, allowing for players without building control to see quasi-instantaneous updates of the *city* as their partner builds.

Both individual rewards (money earned) and team rewards (total money earned) are tracked by Collaborative Math-City. Individual rewards and team rewards are referred to collectively as the *economy*. Reward information is stored and sent over the network connection to synchronize the *economy* between participants. When players answer math questions correctly, their earned money is reflected in the total money available (money earned by the team), as well as their individually earned money. All teammates can view their own contribution as well as their partners' contribution to the team's success (see Figure 2).

4.2 Vocal Communication

Collaborative Math-City uses the user datagram protocol (UDP) to enable vocal communication between collaborative partners. The UDP protocol was selected for vocal communication because delivery latency is prioritized over order and reliability; minimized latency is essential in networked vocal communication. To enable vocal communication, Collaborative Math-City captures sound from the computer's built-in microphone (or externally-connected microphone) and transmits that sound over the UDP socket connection. This vocal communication allows each participant to communicate questions and answers, and work collaboratively to solve problems.

5. Exploratory Study

We conducted an exploratory study to analyze the efficacy of Collaborative Math-City with elementary school students. The purpose of the evaluation was threefold: (a) compare single-player mode performance to multiplayer mode performance, where performance is defined as the number of math questions answered; (b) evaluate how and what type of information students communicate while in multiplayer mode; and (c) evaluate if and how collaboration impacts participation in Math-City, i.e. evaluate if both students in a pair answer math problems and contribute to the city equally.

5.1 Participants and Procedures

We evaluated Collaborative Math-City with two different groups of elementary school students. During the first evaluation two researchers were present and during the second evaluation four researchers were present. The researchers observed students while playing, analyzed interactions amongst students and kept detailed notes on what type of communication occurred, if the students encountered any issues, and if any inequalities in participation arose. The first evaluation was completed with six students and the second evaluation was completed with eight students. For the first evaluation, each researcher observed three students, and for the second evaluation, each researcher observed two students. The two groups of students and corresponding genders and grade-levels are included in Table 1.

Before each evaluation began, one researcher described the Math-City user interface to the students. Students were told how to select elements to build a city, how to bring up the window to attempt to answer math questions, and how to interpret the happiness indicators located at the top of the tool bar. For the evaluation, each student played Math-City for two twenty-minute sessions. Students played Math-City in single-player mode during the first twenty-minute session and Collaborative Math-City with a partner during the second twenty-minute session. For the Collaborative Math-City session, students were paired according to age and, when possible, gender (as to avoid cross-gender discrepancies documented in the literature [10, 11]). Table 1 summarizes the student pairings used for each collaborative session.

Table 1: Exploratory study participant information and student pairings used during the Collaborative Math-City session.

		Gender	Grade			Gender	Grade
Evaluation One	Pair 1	male	5^{th}	Evaluation Two	Pair 4	male	6^{th}
		male	5^{th}			male	6^{th}
	Pair 2	male	5^{th}		Pair 5	male	6^{th}
		male	5^{th}			male	6^{th}
	Pair 3	female	6^{th}		Pair 6	male	5^{th}
		female	6^{th}			female	5^{th}
					Pair 7	female	6^{th}
						female	6^{th}

While students played Collaborative Math-City, the researchers observed the type of communication that occurred between the students, how the students interacted with the game and the turn-taking protocol, and if and how the students collaborated on math questions. The number of math questions attempted and answered correctly was recorded after each 20-minute session (i.e., individual and collaborative sessions). After completion of both 20-minute sessions, participants filled out a short questionnaire.

Table 2: The number of math questions answered correctly over the number attempted. Results are reported for both sessions: single-player (single) and Collaborative Math-City (Collab.).

	Single	Collab.		Single	Collab.
Pair 1	$\frac{4}{4}$ (100%)	$\frac{11}{12}$ (92%)	Pair 4	$\frac{20}{25}$ (80%)	$\frac{25}{48}$ (52%)
	$\frac{1}{1}$ (100%)	$\frac{8}{14}$ (57%)		$\frac{25}{42}$ (59%)	$\frac{16}{20}$ (80%)
Pair 2	$\frac{4}{5}$ (80%)	$\frac{3}{5}$ (60%)	Pair 5	$\frac{4}{4}$ (100%)	$\frac{9}{11}$ (82%)
	$\frac{5}{7}$ (71%)	$\frac{12}{31}$ (39%)		$\frac{7}{12}$ (58%)	$\frac{32}{46}$ (70%)
Pair 3	$\frac{1}{5}$ (20%)	$\frac{6}{13}$ (46%)	Pair 6	$\frac{4}{14}$ (29%)	$\frac{6}{20}$ (30%)
	$\frac{1}{1}$ (100%)	$\frac{10}{18}$ (56%)		$\frac{7}{8}$ (88%)	$\frac{4}{5}$ (80%)
			Pair 7	$\frac{14}{23}$ (61%)	inc.
				$\frac{7}{17}$ (41%)	inc.

5.2 Results

We analyzed results qualitatively and quantitatively. Table 2 displays the number of math questions attempted and the number of math questions answered correctly by each student. *Pair 7* had hardware issues during the collaborative session, and we were unable to collect statistics on mathematical performance (indicated by "inc." in Table 2). In reviewing the data in Table 2, we cannot arrive at a strong conclusion about how collaboration impacts the number of math questions attempted and correctly answered; however, a few observations follow. In general, most students attempted more math problems during the collaborative session than in the individual session. We believe there are two reasons for this fact: 1) the students were learning the interface during the single-player session (i.e., students were more familiar with the game during the collaborative session) and 2) students were working with a partner during the collaborative session, which gave them the ability to answer math questions while someone else (his/her partner) built the city. Additionally, we note that eight of the students correctly answered a higher percentage of math problems while playing in single-player mode, and only four of the students correctly answered a higher percentage of math problems while playing in multiplayer mode. Moreover, two of the researchers noted that students seemed more rushed to answer math problems while playing in multiplayer mode. We believe this could be due to the fact that in multiplayer mode, students have the ability to see how much money they have earned and how much money his/her partner has earned. While this feature was added to the game to prevent participation inequalities, it could be crating a competitive environment in which students feel like they need to quickly answer as many math questions as they can to earn the team money. In general, more comprehensive studies are needed in order to draw conclusions about how collaboration impacts mathematical learning in Collaborative Math-City.

During the Collaborative Math-City session, some of the student pairs talked exclusively about how to build the city (three pairs), while others talked about how to build the city and how to answer math problems (four pairs). Although the game's implementation does not encourage collaboration on math problems, it was common for collaborative pairs to talk about the math problems. For example, a student in *Pair 2* said that he "didn't know how to reduce fractions," and his partner generally described how to reduce fractions. When talking about math problems, exact answers were not communicated. Instead, information on how to solve the problems were provided. We attribute this to the fact that students were working on individual computers and could not see one another's math problems. When one student would ask for help, s/he would phrase the question in general terms, sometimes including and sometimes excluding the actual numbers in the math problem. While more research is required to determine how collaboration on mathematics within Collaborative Math-City impacts learning, we note that Jeong [22] states that if students develop the ability to communicate how to solve problems, a deeper understanding of the underlying methods is gained; thus, we believe that collaboration on math problems in Collaborative Math-City could serve as a tool to strengthen mathematical learning.

While playing Collaborative Math-City, all of the student players discussed city construction. Many collaborative pairs would ask if it was a "good idea" to place certain types of buildings in particular places. Others would ask specific questions about how to make the citizens happier: "How do you solve pollution problems?" Not all of the student pairs, however, were able to effectively use the turn-taking protocol. For example, although the interface was reviewed with the students before the evaluation began, *Pair 2* did not figure out that they both had the ability to contribute to the city until half way through the Collaborative Math-City session. Additionally, *Pair 6* was unable to cooperate and work as a team. The male student in *Pair 6* continually took control of building the city, and the female student did not get a chance to contribute to the city very much. The female student in *Pair 6* voiced frustration about the turn-taking protocol in her student survey, and the researcher who observed while she was playing noted that she was very vocal about how much money she had earned and how her partner was spending all of *her money*. We note that *Pair 6* was our only mixed-gender pair and this type of interaction could be due to the mixed-gender pairing, to the particular pairing of students, or due to an inadequate turn-taking protocol. In short, the *take* turn-taking protocol did not supply an adequate and fair method for both students in every pair to contribute to building the city. To alleviate this participation tension, a new turn-taking protocol could be researched and implemented within the Math-City framework. For example, simply adding a minimum turn-length timer on top of the *take* turn-taking protocol could help alleviate the frustration and troubles that *Pair 6* encountered.

With the exception of *Pair 6*, all of the other student pairs would ask for control ("Can I build now?"), tell their partner they were about to take control ("I'm going to take control now, OK?"), or ask their partner if they were ready to switch when they wished to transfer building control ("Do you want to switch now?"). In general, one student would answer math questions while the other student built the city and then they would switch roles. During the evaluation, we noted that two of the student pairs, *Pair 2* and *Pair 5*, did not equally contribute to the city. In both *Pair 2* and *Pair 5*, one student primarily answered math questions to earn money for the city while the other student spent the majority of the time building the city. This suggests that tracking the amount of money each individual has contributed to the team's success is not sufficient in ensuring both members of the team participate in both Math-City tasks. Our results suggest that within a collaborative electronic environment with more than one primary task, it is important to supply transparency to individual contributions to all tasks. If the interface only reports individual contribution to one task, some of the students may only focus on that one task, leaving the other task for his/her partner to complete. To balance participation on both primary tasks in Collaborative Math-City, in the future we will provide an additional metric for students to compare their success: the percentage of the city they have contributed.

5.2.1 Questionnaire

The post-evaluation questionnaire asked students several questions about playing Collaborative Math-City. The following list details the survey questions:

1) When playing with a partner (circle one):
   a) We worked together very well and built the city together.
   b) We each worked on our own screen and didn't talk about how to build the city very much.
2) When adding buildings to your city (circle one):
   a) We talked about what buildings we wanted to add.
   b) We each just added whatever we wanted.
   c) My partner added buildings without talking to me.
   d) I added buildings without talking to my partner.
3) Do you think the city you created (circle one):
   a) Was better because you were able to work with a partner.
   b) Would have been better if you could have worked alone.
4) Did you and your partner solve math problems together? Yes. No.
5) How did you and your partner work on math problems? (circle one):
   a) I helped my partner learn how to solve the problems.
   b) My partner helped me learn how to solve the problems.
   c) We helped each other solve the problems.
   d) We just shared the answers to the questions.
   e) We did not help each other on math problems.
6) Did you like playing with a partner? Yes. No.
7) Did you get frustrated when your partner took control of building the city? Yes. No.
8) Did you prefer to play the game without a partner? Yes. No.
9) Was it hard to start playing the multi player game? Yes. No.
   a) If it was hard, what was hard about it?
10) Was the game fun? Yes. No.

Figure 3 reports the survey question number and corresponding answers. Overall, the students were more enthusiastic about playing in multiplayer mode than they were about playing in single-player mode. The only student who reported not liking the multiplayer mode was the female student in *Pair 6* (Question 6 and 8). Again, this could be due to a mixed-gender pairing, or be indicative of an unfair turn-taking protocol with the current Collaborative Math-City implementation. Additionally, three students reported that they got frustrated when their partner took control of building the city (Question 7). As we stated earlier in Section 5.2, we believe that implementing a turn-taking protocol that enforces a minimum turn length time-limit could alleviate some of this frustration. In the survey, two of the students reported that it was hard to start playing in multiplayer mode (Question 9). These two students reported that it was hard because: 1) one student couldn't hear their partner at first ("Connecting so we could hear.") and 2) one student was used to playing individually, and calibrating to the multi-player session was hard ("Because you were used to doing everything by yourself in the single player game."). The first concern is (and was) easily fixed by turning up the machine's volume. However, there is no trivial solution to address the second concern.

6. Conclusions and Future Work

This paper presents the design, development, and evaluation of Collaborative Math-City. The design of Collaborative Math-City considers basic pedagogies, as well as educational electronic game and collaborative electronic game research. Our evaluations of Collaborative Math-City concluded that Math-City is appropriate for collaboration and that in general, students prefer to play Collaborative Math-City over

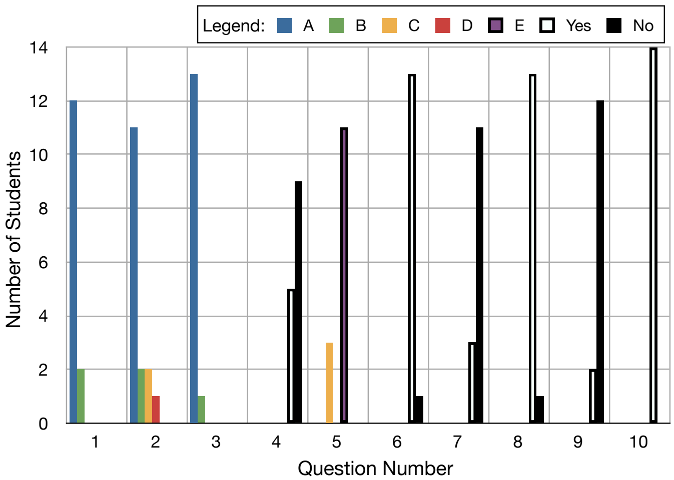


Figure 3: Student's answers to the post-evaluation survey. The students in *Pair 6* both supplied two answers to Question 2.

Math-City. For the purpose of this research, Collaborative Math-City was implemented to enable a two-player collaborative environment. This work contributes to research on collaborative electronic educational environments and implements a framework for communication and synchronization. In the future, we plan to extend the Collaborative Math-City framework to support 2 – 4 participants per team. We will use this extension to Collaborative Math-City to research and analyze how a larger collaborative team impacts electronic educational environments.

As mentioned in Section 5.2, some students did not contribute as much as their partners for the two main tasks: building the city and answering math questions. We plan to address these participation inequalities by extending Collaborative Math-City to track individual contributions to the city construction in addition to individual contributions to the team's pool of money. We hypothesize that providing students with transparency to how much they have contributed to the city will provide a compelling reason to assist in constructing the city. Additionally, we found that the *take* turn-taking protocol does not always provide student pairs with equal ability to contribute to the construction of the city. To address this issue, we will enhance the *take* turn-taking protocol, perhaps with a minimum turn timer, to create a more fair method to transfer building control between students.

Acknowledgements

The authors would like to thank Chris Walsh for his support and assistance during both evaluation sessions. We also thank Chris Navrides, Jeff Neef, Chad Kembel, Eric Chiu, Chris Poupore, Ricky Walker, Joe Zeimen, Amanreal Bajwa, Nick Hansen, and Levente Sipeki for their work on the Math-City implementation.

References

[1] J. L. McGrenere, "Design: Educational electronic multi-player games *A Literature Review*," Ph.D. dissertation, The University of British Columbia, 1996.

[2] K. Inkpen, K. S. Booth, S. D. Gribble, and M. Klawe, "Give and take: Children collaborating on one computer," in *CHI '95: Human Factors in Computing Systems*. ACM Press, 1995, pp. 258–259.

[3] D. Africano, S. Berg, K. Lindbergh, P. Lundholm, F. Nilbrink, and A. Persson, "Designing tangible interfaces for children's collaboration," in *CHI '04 extended abstracts on Human factors in computing systems*, ser. CHI '04, New York, NY, USA, 2004, pp. 853–868.

[4] M. Virvou, G. Katsionis, C. Manos, and K. Tourtoglou, "VR-ENGAGE: A Virtual Reality Educational Game that Incorporates Intelligence," in *Proceedings of IEEE International Conference on Advanced Learning Technologies*, 2002.

[5] E. Klopfer, J. Perry, K. Squire, M.-F. Jan, and C. Steinkuehler, "Mystery at the museum: a collaborative game for museum education," in *Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!*, ser. CSCL '05, 2005, pp. 316–320.

[6] K. Inkpen, J. McGrenere, K. S. Booth, and M. Klawe, "The Effect of Turn-Taking Protocols on Children's Learning in Mouse-Driven Collaborative Environments," in *Proceedings of the conference on Graphics interface '97*, Toronto, Ont., Canada, Canada, 1997, pp. 138–145.

[7] K. Inkpen, J. Mcgrenere, K. S. Booth, and M. Klawe, "Turn-Taking Protocols for Mouse-Driven Collaborative Environments," 1997.

[8] P. R. Jermann, "Computer Support for Interaction Regulation in Collaborative Problem-Solving," Ph.D. dissertation, University of Geneve, 2004.

[9] M. Kapur, J. Voiklis, and C. K. Kinzer, "Sensitivities to early exchange in synchronous computer-supported collaborative learning (cscl) groups," *Computer Education*, vol. 51, pp. 54–66, August 2008.

[10] S. D. Scott, R. L. Mandryk, and K. M. Inkpen, "Understanding Children's Collaborative Interactions in Shared Environments," *Journal of Computer Assisted Learning*, vol. 19, pp. 220–228, 2003.

[11] K. Inkpen, K. S. Booth, M. Klawe, and R. Upitis, "Playing together beats playing apart, especially for girls," in *The first international conference on Computer support for collaborative learning*, ser. CSCL '95, Hillsdale, NJ, USA, 1995, pp. 177–181.

[12] P. Moreno-Ger, D. Burgos, I. Martínez-Ortiz, J. L. Sierra, and B. Fernández-Manjón, "Educational game design for online education," *Comput. Hum. Behav.*, vol. 24, pp. 2530–2540, September 2008.

[13] I. Polycarpou, J. Krause, C. Rader, C. Poupore, and E. Chiu, "Math-City: An educational game for K-12 mathematics," in *In the Proceedings of the 2010 World Conference on Learning, Teaching, and Administration (WC-LTA). Social and Behavior Sciences*, 2010, pp. 845 – 850.

[14] E. G. Cohen, "Restructuring the Classroom: Conditions for Productive Small Groups," *Review of Educational Research*, vol. 64, pp. 1–35, 1994.

[15] S. Hymel, B. Zinck, and E. Ditner, "Cooperation Versus Competition in the Classroom," *Exceptionally Education Canada*, vol. 3, pp. 103–128, 1993.

[16] E. Zakaria, L. C. Chin, and M. Y. Daud, "The Effects of Cooperative Learning on Students Mathematics Achievement and Attitude towards Mathematics," *Journal of Social Sciences*, vol. 6, pp. 272–275, 2010.

[17] C. E. Gilbert, "The effects of cooperative learning and teaming on student achievement in elementary mathematics," Ph.D. dissertation, TUI University, 2008.

[18] S. G. Harkins and K. Szymanski, "Social loafing and self-evaluation with an objective standard," *Journal of Experimental Social Psychology*, vol. 24, no. 4, pp. 354 – 365, 1988.

[19] J. A. Shepperd, "Productivity Loss in Performance Groups: A Motivation Analysis," *Psychological Bulletin*, vol. 113, no. 1, pp. 67 – 81, 1993.

[20] J. Hereford and W. Winn, "Non-speech Sound in Human-Computer Interaction: A Review and Design Guidelines," 1994.

[21] S. Greenberg and R. Bohnet, "GroupSketch: A Multi-User Sketchpad for Geopraphically-Distributed Small Groups," in *Graphics Interface '91 Conference Proceedings*, 1991, pp. 207–215.

[22] H. Jeong, "Knowledge co-construction during collaborative learning," Ph.D. dissertation, University of Pittsburgh, PA, 1998.

A web-based system for connecting undergraduate students and researchers via textbooks for multi-disciplinary, research-oriented education

Steven B. Kraines[1,*], Weisen Guo[1], and Takaki Makino[1]
[1]Science Integration Programme – Human, Division of Project Coordination, the University of Tokyo, 5-1-5 Kashiwa-no-ha, Kashiwa-shi, Chiba Pref. 277-8568 Japan
[*]sk@scint.dpc.u-tokyo.ac.jp

Abstract – *An application of the University on Textbooks (UoT) system to a web-based platform for supporting organic growth of an knowledge network related to sustainability science is described. The UoT system, which aims to help university students learn about cutting-edge research at their university through the medium of textbooks, uses a description logics reasoner to generate semantic links between ontology-based descriptors of topics in textbooks and of research articles. Those semantic links enable students to navigate between well-structured introductory materials provided by textbooks and advanced research on specific topics presented in research articles. The new UoT system maps sections from the book "Vision 2050: A Roadmap for a Sustainable Earth" to research articles and web pages of researchers at the University of Tokyo studying issues related to sustainability, energy and materials. The system is described, and implications for supporting education in highly multidisciplinary subjects such as sustainability science are discussed.*

Keywords: Expert knowledge, knowledge network, logical inference, semantic matching, undergraduate education, sustainability science

1 Introduction

Research universities are mandated to both 1) provide world-class undergraduate education and 2) advance world-class research. In theory these two tasks could work together, e.g. by effectively communicating knowledge produced through university research to students via the education system. However, in reality effective communication between researchers and students is rare, mainly because of the way in which researchers publish their knowledge in research articles. We identify three major obstacles preventing undergraduate students from accessing advanced research at their universities. First, undergraduate students do not know how to search for research articles in the literature [17]. Second, undergraduate students do not have the background knowledge needed to read and understand research articles [36]. Third, undergraduate students do not have sufficient knowledge breadth to understand how knowledge in a single research article connects with the overall knowledge in the field [30].

In order to overcome these obstacles, the research articles need to be arranged or "structured" in some way so that undergraduate students can easily access and learn about them within the context of their normal educational activities. We suggest that textbooks could be used to provide this structure. Textbooks are comprehensive overviews of specific fields in a well-structured form, which make textbooks well-suited as platforms from which students can explore research articles and other knowledge resources. The structure that is provided by textbooks, which enables students and teachers to share a broad common understanding of a field of science, is a consequence of the relatively slower rate in which the content of textbooks are updated in comparison to the research literature. We suggest that by using semantic matching techniques to position research articles in the knowledge structure provided by the textbooks, those articles could be made more accessible to students. In this sense, the textbook is like a slowly growing tree upon whose branches researchers hang their new research findings.

As an implementation of this idea, the University on Textbooks system has been developed that links research articles to a textbook in the domain of life sciences. Descriptions in a formal knowledge representation language grounded in a description logic, which we call semantic statements, were created for 100 passages from a life sciences textbook used by second-year undergraduates at the University of Tokyo and for 400 research articles in life sciences authored by researchers at the University of Tokyo [26]. The University on Textbook system is comprised of a web-accessible textbook browser, which we call the UoT textbook browser, and a web-based semantic statement editing and matching system called EKOSS for Expert Knowledge Ontology-based Semantic Search. The UoT textbook browser, which is directly accessible by a web browser with no need for plug-ins or any other software installation, provides a paginated and customizable layout showing the text and figures from the textbook. Pagination and layout control is believed to be important for making textbooks easy to read on a web browser [4], [45]. The UoT textbook browser and the EKOSS system are connected via a set of web services that pass information about which researchers have written research articles that are semantically related to passages from the textbook, as soon as that information is updated.

Here we describe a new University on Textbooks system that we have developed, which uses the book "Vision 2050: A Roadmap for a Sustainable Earth" to provide a foundation for accessing advanced research related to sustainability science. The contents of the book have been adapted to the textbook browser, and we have created semantic statements to describe about 70 specific passages from the book varying in length from one sentence to a few paragraphs. Using a semantic matching algorithm developed in previous work, we evaluate the semantic similarity between each textbook statement and about 200 semantic statements that we created based on research articles and web pages written by researchers at the University of Tokyo to report research work related to the topics of sustainability, energy and materials. We describe the system and discuss the effectiveness for supporting undergraduate education in the highly multidisciplinary field of sustainability science.

2 Background and Related Research

Most recent work in e-learning and web-based education is focused on "stand-alone" modules of interactive learning material. These e-learning modules can be helpful supplementary teaching tools; however, they do not have sufficient structure and connection to provide students with a comprehensive learning environment. Even highly structured web resources, such as Wikipedia, lack the clarity and integrity that is needed in an undergraduate course. Textbooks, on the other hand, contain knowledge of the target domain that has been carefully structured, often through many revisions [1], making them suitable media for introducing the basic concepts and tools in specific academic fields to students who have very little background knowledge. Moreover, textbooks, particularly those that have been around for awhile, are often used as references by practicing researchers and teachers, e.g. to recall previously learned but partially forgotten knowledge.

The use of books as a framework for organizing knowledge resources in a particular field is certainly not a new idea [43], [33]. Many publishers provide web-accessible supplemental content for textbooks, such as problem sets, graphics, simple software applications, multimedia materials, and even web-accessible textbook readers giving free or prepaid access to the actual textbook (e.g. [40]). With web-based textbook readers, this supplemental content can be provided to readers "on the fly" [10], [4]. And books on the web can also support movies, animations, three dimensional graphics, and interactive learning applications [31], [32], [30].

The problem of how to map resources to specific sections of a textbook is more difficult. Some researchers have tried to map parts of a textbook to research articles or even other parts of the same textbook using glossary terms [4], [43]. However, the accuracy is limited by the ability of computer to match the glossary terms to the research articles, which is usually done in a typical "bag-of-words" fashion. Although other researchers have tried to use natural language processing to extract relationships between glossary terms [13], semantically rich relationships cannot be extracted reliably due to ambiguities of natural language.

The task of effectively matching knowledge resources has received considerable attention recently [24], [38], [12], [46]. Matching techniques can be differentiated into automatic approaches and manual approaches. Automatic approaches include semantic methods from simple keyword indexing to more sophisticated natural language understanding, and statistical methods, e.g. the use of collaborative filtering or inverse document term frequency measures. Manual approaches involve a human in the matching process, often to provide some form of computer-interpretable descriptors, which range from simple lists of keywords to descriptive statements in some formalized knowledge representation knowledge [35]. The computer-interpretable descriptors can then used by some computational algorithm for matching, as well as for classifying resources and extracting information of potential interest [18]. Semi-automatic methods that use automatic natural language processing techniques to facilitate manual creation of computer-interpretable descriptors have been proposed [20].

We further contrast different type of manual approaches 1) by the form of the descriptors created to represent the knowledge resources and 2) by the person who creates those descriptors. Most existing approaches utilizing descriptors that are more sophisticated than simple tag lists depend on expert knowledge curators to create those descriptors. However, it has been suggested that currently available web technologies could be used to "wrap" computer-interpretable knowledge representation languages with intuitive user-friendly authoring tools [11]. The ontologies that have emerged in the context of the Semantic Web could be used as formal knowledge representation languages for humans to author descriptors that enable computers to act as more effective and "intelligent" matchmakers for exchanging and integrating different forms of scientific knowledge [6], [44], [41], [29], [2], [15]. This could enable the original knowledge creators to author their own computer-interpretable descriptors in a reliable and semantically rich manner. And if computer interpretable descriptors of knowledge resources were provided by the original human creators, then it should be possible to achieve a more effective and accurate forms of computer-aided knowledge sharing [19], [42], [7], [14], [37], [3], [8], [9]. Engaging the entire research community in the creation of computer-understandable descriptors is a scalable solution to the problem of curating all of the research articles related to a particular knowledge domain such as sustainability science [5], [16], [11]. To make this kind of "crowd sourcing" happen, researchers would have to be provided with some form of incentive for creating their own computer-understandable descriptors, perhaps by making their creation a part of the process of submitting research articles or grant proposals.

3 Methods

The basic concept of connecting students to researchers at their university via textbooks is shown in figure 1. Semantic statements are created both for passages from the textbook and for research articles and other shared resources created by researchers to express their expert knowledge. The EKOSS semantic matching algorithm is then used to create semantic links between the textbook statements and the researcher statements, as shown by black curved arrows (figure 1). The links can be traversed in either direction, enabling a person reading a research article that has been linked to a textbook to refer to basic knowledge in the textbook related to the topics mentioned in the article.

For our application of the University on Textbook system to the domain of sustainability science, we have chosen the textbook "Vision 2050: A Roadmap for a Sustainable Earth" [23]. We selected this textbook in part because it is freely accessible from SpringerLink and in part because it covers many of the topics of research that are the focus of researchers at the University of Tokyo. The book, which is written in English, has 162 pages, 8 chapters, and 33 black and white figures. We created semantic statements for about 70 passages from the book ranging from one sentence to several paragraphs in length. We then determined the semantic similarity between those semantic statements and about 200 semantic statements describing research articles and web pages by researchers studying issues related to sustainability science. We calculated the similarities by using a semantic matching algorithm and scoring system developed in previous work, which utilizes logic and rule based inference together with inverse term and document frequencies [22].

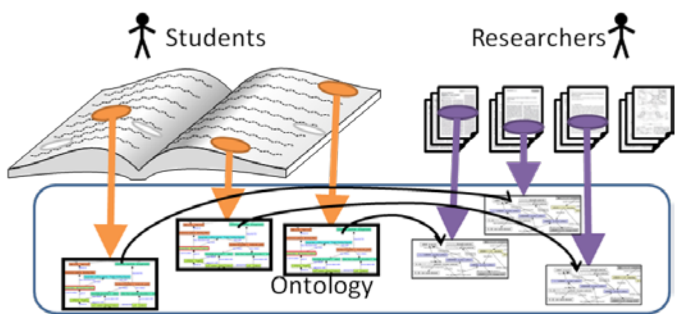


Figure 1: Bridging the knowledge gap between undergraduate students and university researchers by using ontologies to link textbooks and research articles

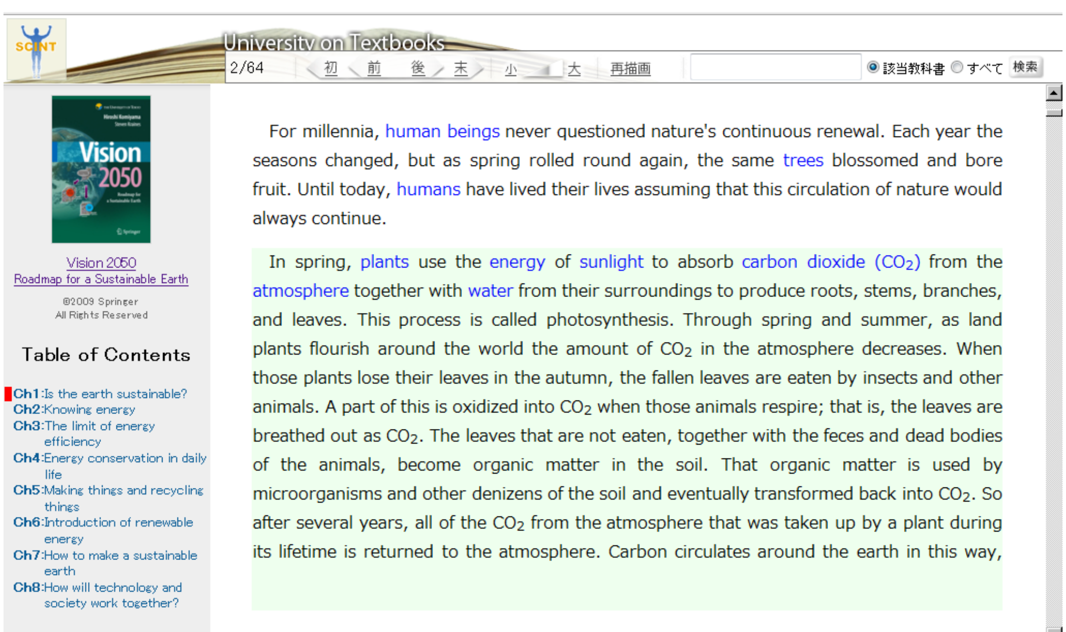


Figure 2: A view of the UoT Textbook Browser with a large font size selected showing the passage from chapter 1 of "Vision 2050". Terms in blue font are keywords that have been mapped to classes from the ontology. The light green background indicates that the passage has an associated semantic statement.

4 Results

The University on Textbook system uses a web-based multi-column paginating textbook browser, which we call the UoT Textbook Browser, to provide the user interface for reading the selected textbook and accessing researchers via the semantic links. The text flow control, search features, and popup information windows are all implemented using Javascript, so that the UoT Textbook Browser can function on standard web browsers. Examples of the basic UoT Textbook Browser views are shown in figures 2 and 3.

The UoT textbook browser supports layout arrangement, text justification and hyphenation, handling of dangling punctuations, and a pop-up window management system for showing information on ontology classes and researchers that are obtained via web services provided by the EKOSS system. The EKOSS system provides two main web services for giving information on classes and researchers to the UoT textbook browser. The first web service provides information about a class from the ontology, including a short definition of the class, the number of times the class has been used in the semantic statements handled by the EKOSS system, the number of researchers using that class, the number of research articles in which instances of that class appear, and contact and affiliation information for each researcher who is an author of one of the research articles. The last type of information is helpful for assisting undergraduate students in finding thesis advisors or graduate school laboratories to which to apply. An example is shown in figure 4.

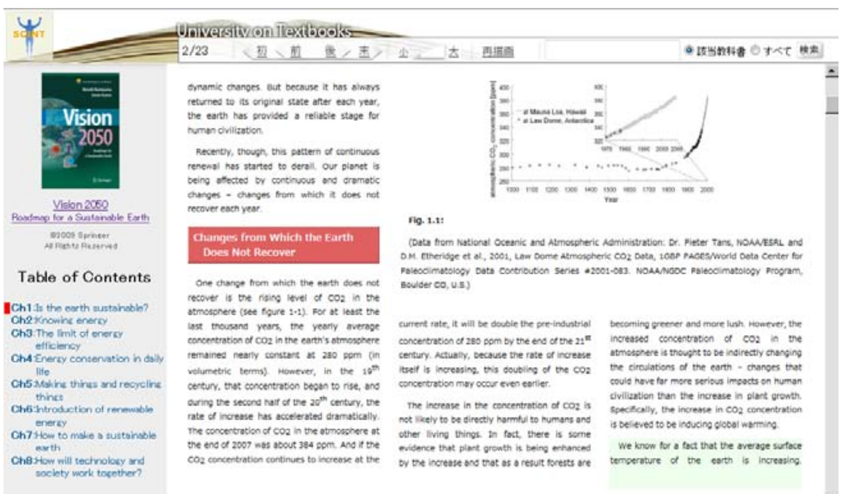


Figure 3: A view of the UoT Textbook Browser with a small font size selected. The textbook browser automatically repaginates to multiple columns and repositions figures.

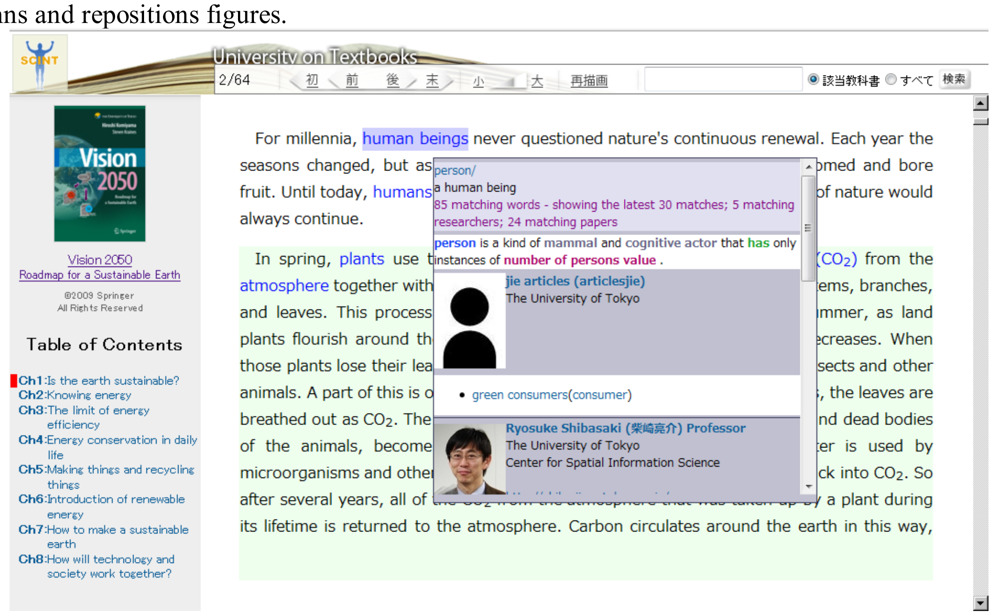


Figure 4: Popup window showing information provided by the first EKOSS web service in response to a query for a class from the ontology. A short definition and usage statistics of the class are given, followed by a list of researchers using the class or a subclass in their research articles. The researcher information includes links to a researcher information page giving contact information and lists of semantic statements, the EKOSS ontology class browser, and the actual semantic statement for the research article.

Figure 5: View of the UoT Textbook Browser showing the popup window containing information in three tags that is provided by the second EKOSS web service for a semantic statement describing the highlighted passage in the textbook. The tab that is shown lists the researchers who have written research articles whose semantic statements are semantically similar to the textbook statement. Each researcher description includes links to 1) a researcher information page giving contact information and lists of semantic statements, 2) the matching research article, and 3) the research article's semantic statement. The "subject verb object" triples from the research article statement that matched with the textbook statement are rendered in natural language.

The second EKOSS web service, shown in figure 5, provides information and semantically linked research articles for a passage from the textbook, for which a semantic statement has been created. The information includes an image file of the graph representation of the semantic statement and a formatted natural language representation generated by the EKOSS NLG algorithm [25]. The semantically linked articles are those that have semantic statements with sufficient semantic similarity to the textbook semantic statement, as calculated by the EKOSS semantic matching algorithm. The results of the semantic matching (the tab that is currently shown in figure 5) are given as a list of researchers who have authored one of the matched semantic statements. For each matching researcher, the web service provides affiliation information, the name and link of the matching research article and its semantic statement, and a rendering in natural language of the "subject verb object" triples from the semantic statement that were found to match with the textbook statement.

5 Discussion

There are a number of possible applications of the University on Textbook system in undergraduate education. The driving application for us has been to help second year undergraduate students decide which department they would like to join, to help third year students decide with which specific research professors they would like to do their undergraduate thesis work, and to help fourth year students select laboratories in which they would like to do their graduate study. For that reason, we have focused on providing information that enables students to access the research faculty at the university who have written research articles mapped to a particular passage from the textbook. However, because the UoT textbook browser is implemented in standard Javascript, it is possible to extend its functionality by integrating different multi-media, semantic web and web 2.0 technologies. For example, support could be added for multimedia such as animated movies, 3D images, and interactive "game" applications (e.g. [34]). Also, because it is possible to annotate text and use hyperlinks freely, social network system technologies could be used to provide interactive and user-customization features such as annotation, bookmarking, and "just in time teaching" mechanisms, e.g. to provide a feedback mechanism for the students to post questions or comments [30].

This linking of "person to person" is also in line with the underlying goal of the University on Textbook project that we touched on in the introduction, which is to motivate researchers to create their own semantic statements describing their research. The notion is that although research faculty at universities are eager to attract students to their laboratories, they are often overwhelmed by research-oriented tasks and do not have time to communicate with the undergraduate students directly. The University on Textbook system enables those faculty members to use their regular research publications as advertising material to the undergraduate student population for the (ideally small) added cost of creating a semantic statement. The semantic statement could then be used in other “intelligent” computer services, such as semantic search [28], knowledge mining [27], natural language generation [25], recommender systems [21] and matching with peer reviewers.

In future work, we will attempt to increase the number of semantic statements describing research articles to link with the University on Textbook system. A critical part of this work is the development of authoring tools that utilize natural language processing techniques to help researchers to create semantic statements describing their research publications easily and accurately. We also plan to add additional textbooks to the University on Textbook system from life sciences, sustainability science, and other domains of science. Finally, we are planning a study to evaluate the effectiveness of the system in helping students at the University of Tokyo identify research faculty with whom they would like to do their undergraduate thesis and graduate research work.

6 Acknowledgements

Funding support for this research was provided by the President’s Office of the University of Tokyo and the Japan Office of the Alliance for Global Sustainability. The UoT textbook browser was implemented by NalaPro Technologies.

7 References

[1] Alberts, B. (2008): Shortcuts to medical progress? Science 319(5871), 1733.

[2] Allenby, B. (2006): The ontologies of industrial ecology? Progress in Industrial Ecology. An International Journal 3(1): 28-40.

[3] Androutsopoulos, I., Oberlander, J., and Karkaletsis, V. (2007): Source authoring for multilingual generation of personalised object descriptions. Natural Language Engineering, vol 13(3) pp. 191-233

[4] Appleton, L. (2005): Using electronic textbooks: promoting, placing and embedding. The Electronic Library 23(1), 54–63.

[5] Attwood, T.K., Kell, D.B., McDermott, P., Marsh, J., Pettifer, S.R., Thorne, D. (2009): Calling International Rescue: knowledge lost in literature and data landslide! Biochemical Journal vol 242, pp. 317-333

[6] Berners-Lee, T. and Hendler, J. (2001): Publishing on the Semantic Web. Nature, 410: 1023—1024.

[7] Biller, O., Elhadad, M., Netzer, Y., Univerisy, B. G. (2005): Interactive authoring of logical forms for multilingual generation. In Proc 10th European Workshop on Natural Language Generation, pp. 24-31

[8] Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., Berlin F. U., and Eckstein, R. (2005): The Impact of Semantic Web Technologies on Job Recruitment Processes. In Proceedings of the 7th International Conference Wirtschaftsinformatik, pp. 1367-1383

[9] Bradley, J. (2008): Pliny: A model for digital support of scholarship. Journal of Digital Information, 9(1)

[10] Brusilovsky, P., Chavan, G., and Farzan, R. (2004): Social adaptive navigation support for open corpus electronic textbooks. Adaptive Hypermedia and Adaptive Web-Based Systems 3137, 24-33.

[11] Ceol, A., Chatr-Aryamontri, A., Licata, L., and Cesareni, G., (2008): Linking Entries in Protein Interaction Database to Structured Text: the FEBS Letters Experiment. FEBS letters, 582(8), 1171-1177.

[12] Chua, A.Y.K., Lam, W., and Majid, S. (2006): Knowledge reuse in action: the case of CALL. J. Information Science 32(3), 251-260.

[13] Crestani, F., and Melucci, M. (2003): Automatic construction of hypertexts for self-referencing: the hyper-textbook project. Information Systems 28(7), 769–790.

[14] Davis, B., Iqbal, A. A., Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., and Handschuh, S. (2008): RoundTrip Ontology Authoring. Intl Semantic Web Conf, ed. Amit P. Sheth et al., LNCS vol 5318 pp. 50-65

[15] Davis, C., Nikolic, I. and Dijkema, G. P. J. (2009): Integration of Life Cycle Assessment Into Agent-Based Modeling. Journal of Industrial Ecology, 13: 306-325.

[16] DeRose, P., Shen, W., Chen, F., Doan, A., and Ramakrishnan, R. (2007): Building structured web community portals: a top-down, compositional, and incremental approach. In VLDB '07: Proc 33rd Intl Conf on Very Large Data Bases, Vienna, Austria, pp. 399-410.

[17] Flaspohler, M. R., Rux, E. M., and Flaspohler, J. A. (2007): The annotated bibliography and citation behavior: enhancing student scholarship in an undergraduate biology course. CBE Life Sci. Educ. 6, 350-360.

[18] Garfield, E. (2001): A retrospective and prospective view of information retrieval and artificial intelligence in the 21st century. JASIST 52 (1), 18-21.

[19] Gerstein, M., Seringhaus, M. and Fields, S. (2007): Structured digital abstract makes text mining easy. Nature, 447: 142.

[20] Goldstein-Stewart, J. and Winder, R.K. (2009): Designing a system for semi-automatic population of knowledge bases from unstructured text. In Proceedings of the Intl Conf of Knowledge Discovery and Information Retrieval (KDIR 2009) Madeira, Portugal, 2009. pp. 88-99

[21] Guo, W. and Kraines S.B. (2010): Semantic content-based recommendations using semantic graphs, Arabnia, Hamid R. (ed.) Advances in Comp. Biology, Advances in Experimental Medicine and Biology, 2010, Vol. 680, Part 7, pp: 653-659, Springer Science+Business Media, LLC 2010.

[22] Guo, W. and Kraines S.B. (2008): Explicit scientific knowledge comparison based on semantic description matching. ASIST Annual meeting 2008, Columbus, Ohio.

[23] Komiyama, H. and Kraines, S.B. (2008): Vision 2050: A Roadmap for a Sustainable Earth. Springer.

[24] Kostoff, R.N. (2002): Overcoming specialization. BioScience 52(10), 937-941(5).

[25] Kraines, S.B. and Guo, W. (2009): Using Human Authored Description Logics ABoxes as Concept Models for Natural Language Generation, ASIST 2009 Annual Meeting, Vancouver, British Columbia, Canada.

[26] Kraines, S.B., Makino, T., Guo, W., Mizutani, H. and Takagi, T. (2010): Bridging the Knowledge Gap between Research and Education through Textbooks, In: X. Luo et al. (Eds.): ICWL 2010, LNCS 6483, pp. 121-130.

[27] Kraines, S.B., Guo, W., Hoshiyama, D., Mizutani, H., Takagi, T. (2010): Generating Literature-based Knowledge Discoveries in Life Sciences Using Relationship Associations, Intl Conf. on Knowledge Discovery and Information Retrieval, KDIR 2010, Valencia, Spain, pp: 35-44.

[28] Kraines, S.B., Guo, W., Kemper, B. and Nakamura, Y. (2006): EKOSS: A knowledge-user centered approach to knowledge sharing, discovery, and integration on the Semantic Web. LNCS, Vol 4273, ISWC 2006: 833-2091.

[29] Kraines, S.B., Batres, R., Koyama, M., Wallace, D.R. and Komiyama, H. (2005): Internet-based integrated environmental assessment: using ontologies to share computational models. J. Industrial Ecology, 9: 31-50.

[30] Marrs, K. A., and Novak G. (2004): Just-in-time teaching in biology: creating an active learner classroom using the Internet. Cell Biol. Educ. 3, 49-61.

[31] Mayer, R. E., and Moreno, R. (2002): Animation as an aid to multimedia learning. Educ. Psychol. Rev. 14(1), 87-99.

[32] McClean, P., Johnson, C., Rogers, R., Daniels, L., Reber, J., Slator, B. M., Terpstra, J., and White, A. (2005): Molecular and cellular biology animations: development and impact on student learning. Cell Biol. Educ. 4, 169-179.

[33] McFall, R. (2005): Electronic textbooks that transform how textbooks are used, The Electronic Library 23(1), 72-81.

[34] Miller, L., Moreno, J., Willcockson, I., Smith, D., and Mayes, J. (2006): An online, interactive approach to teaching neuroscience to adolescents. CBE Life Sci. Educ. 5, 137-143.

[35] Mons, B., Ashburner, M., Chichester, C., van Mulligen, E., Weeber, M., den Dunnen, J., van Ommen, G.J., Musen, M., Cockerill, M., Hermjakob, H., Mons, A., Packer, A., Pacheco, R., Lewis, S., Berkeley, A., Melton, W., Barris, N., Wales, J., Meijssen, G., Moeller, E., Roes, P.J., Borner, K., and Bairoch, A. (2008): Calling on a million minds for community annotation in WikiProteins. Genome Biol. 9(5), R89.

[36] Porter, J. R. (2005): Information literacy in biology education: an example from an advanced cell biology course. Cell Biol. Educ. 4, 335-343.

[37] Power, R. (2009): Towards a generation-based semantic web authoring tool. In Proc 12th European Workshop on Natural Language Generation, Athens, Greece, pp. 9-15

[38] Ruttenberg, A., Clark, T., Bug, W., Samwald, M., Bodenreider, O., Chen, H., Doherty, D., Forsberg, K., Gao, Y., Kashyap, V., Kinoshita, J., Luciano, J., Marshall, M.S., Ogbuji, C., Rees, J., Stephens, S., Wong, G.T., Wu, E., Zaccagnini, D., Hongsermeier, T., Neumann, E., Herman, I., and Cheung, K.-H. (2007): Advancing translational research with the Semantic Web, BMC Bioinformatics 8(Suppl 3), S2.

[39] Razmerita, L., Firantas, R., and Jusevicius, M. (2009): Towards a new generation of social networks: merging social web with Semantic Web. 4th AIS SigPrag Intl Pragmatic Web Conf. Track. http://i-semantics.tugraz.at/2009/papers/owards_a_new_generation_of_social_networks.pdf

[40] Safari Books Online (2009): Safari Books Online – Home. Available at http://www.safaribooksonline.com.

[41] Shotton, D., Portwin, K., Klyne, G., and Miles, A. (2009): Adventures in Semantic Publishing: Exemplar Semantic Enhancements of a Research Article. PLoS Comput Biol, Vol. 5 pp. e1000361

[42] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. and Ciravegna, F. (2006): Semantic annotation for knowledge management: requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web, 4 (1): 14—28.

[43] Vassiliou, M., and Rowley, J. (2008): Progressing the definition of "e-book". Library Hi Tech 26(3), 355-368.

[44] Wang, X., Gorlitsky, R. and Almeida, J. S. (2005): From XML to RDF: how semantic web technologies will change the design of 'omic' standards. Nature Biotechnology, 23 (9): 1099-1103.

[45] Wilson, R., Landoni, M., and Gibb, F. (2003): The WEB Book experiments in electronic textbook design. Journal of Documentation 59(4), 454-477.

[46] Weikum, G., Kasneci, G., Ramanath, M., Suchanek, F. (2009): Database and Information-retrieval Methods for Knowledge Discovery. Comm. of the ACM, 4, 56-64.

An Educational Logic Tool for the Formal Verification, Design by Contract, and Algorithmic Debugging of Imperative Programs

Rafael del Vado Vírseda
Eva Pilar Orna Ruiz, Eduardo Berbis González, Saúl de León Guerrero

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid
Facultad de Informática, Madrid, Spain
rdelvado@sip.ucm.es
{evapilar.orna,eberbis,sauldeleonguerrero}@gmail.com

Abstract— *While logic plays an important role in several areas of Computer Science (CS) and Software Engineering (SE), most of the educational technology developed for teaching logic ignores their application in a more large portion of the CS and SE education domain. In this paper we describe an innovative methodology based on a prototype logic teaching tool on semantic tableaux to prepare and train the students for using logic as a formal proof technique in other topics of CS and SE, such as the formal verification of algorithms and the declarative debugging of imperative programs, which are at the basis of a good development of software. Along the academic course 2009/2010 we have evaluated the part of the tool concerning the analysis of algorithms and software engineering. The tool evaluation is carried by means of tests, some of them managed from a Virtual Campus, with open access to all the students, and another one managed in a computer laboratory with a controlled group. We show this educational experience about the benefits that the students get from the tool in the current teaching of the design of software.*

Keywords: Software Engineering Education, Computers for Education, Formal Verification, Algorithmic Debugging

1. Introduction

Computer Science programs often teach a first year undergraduate course on mathematical logic. The syllabus of the course usually includes syntax and semantics of propositional and predicate logic, as well as some proof systems such as natural deduction, resolution, and semantic tableaux. In some cases, there is also some lecture devoted to explain basic concepts on logic programming and practical work using a *Prolog* interpreter.

Most students find the high degree of rigour required in the learning of these contents daunting. In order to provide learning support to our students, proof visualization tools are always helpful. Many tools for teaching logic have been developed in the last two decades (see `http://www.ucalgary.ca/aslcle/logic-courseware`). Most of these tools concern the construction of proofs in a formal logic using *semantic tableaux*. A semantic tableau [3] is a semantic but systematic method of finding a model of a given set of formulas Γ. A semantic tableau is a refutation system in the sense that a theorem φ is proved from Γ by getting its negation $\Gamma \vdash \neg\varphi$.

While logic plays an important role in several areas of Computer Science and Software Engineering, most of the didactic software developed for teaching logic ignores the application of logic in other topics of Computer Science and Software Engineering Education. We believe that our students could obtain more benefits from the techniques they learn with these tools, thanks to the possibility of applying them in a variety of contexts in advances courses. Moreover, it is frequent that a tool is useful not only to the original purpose it was created for, but also for other subsidiary (though not less important) uses. Hence, there is a need for a prototype tool allowing experiments on teaching logic in a more large portion of the Computer Science and Software Engineering education domain, where the language and the implementation should be accessible enough and popular to ensure that they will be used into the future, and that they remain available in other courses. These motivated us to write this paper.

The aim of this work is to describe an innovative methodology based on a logic teaching tool on sematic tableaux, called TABLEAUX, to prepare our students for using logic as a formal proof tool in other areas of Computer Science, with a special emphasis on the design of algorithms and Software Engineering. Good algorithm design is crucial for the performance of all software systems. For this reason, an ability to create and understand formal proofs is essential for correct program development.

The major contribution of this paper is the development of new tableau methods that give semantically rich feedback to our students for the formal verification and the algorithmic debugging of programs. In this sense, TABLEAUX shows to be a good tool for (a little more) advances students, whose logical skills go beyond the rudiments that the user-level

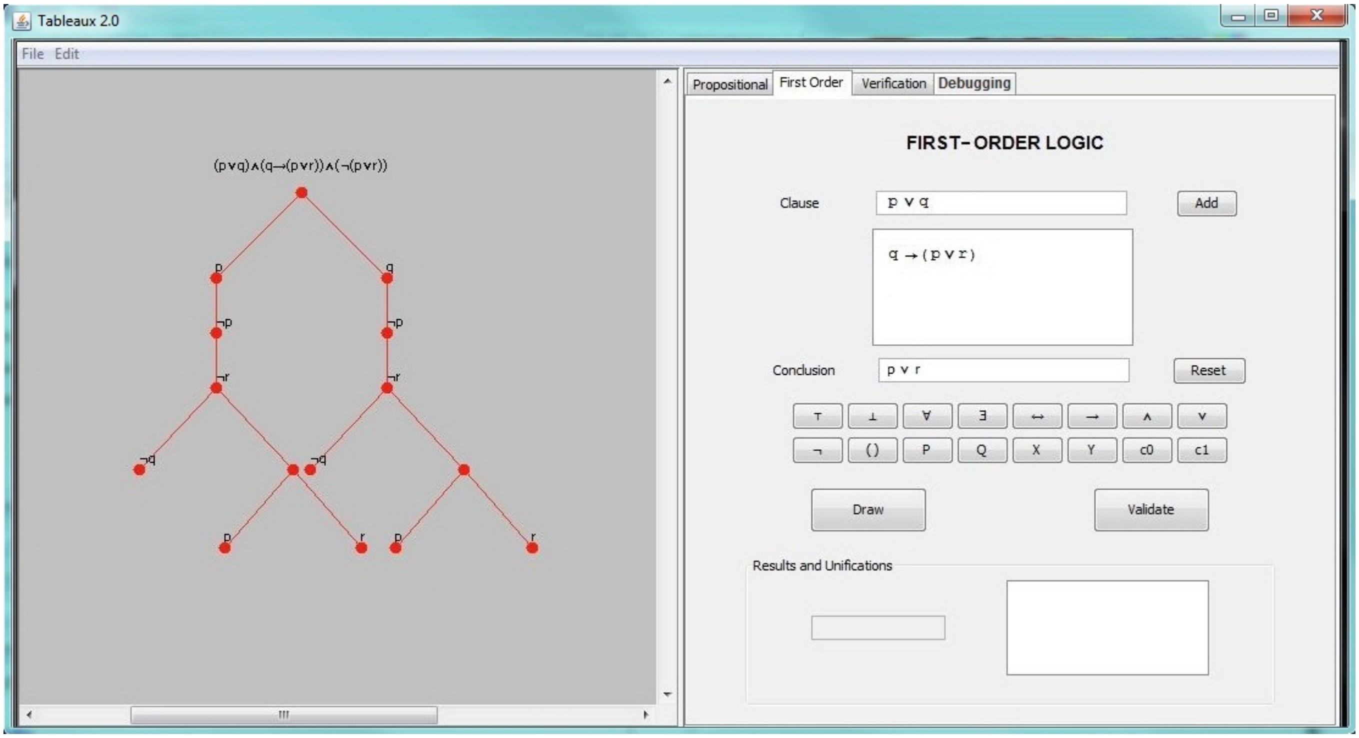


Fig. 1: The logic teaching tool `TABLEAUX` for propositional and first-order semantic tableaux.

interaction with other logic teaching tools can develop. For instance, our tool is used for logic-based methodologies, such as program derivation, reasoning from specifications and assertions, loop invariants, bound functions, etc. This includes topics in areas whose skills and concepts are essential to programming practice independent of the underlying paradigm, as the analysis and the design of correct and efficient algorithms [4].

2. The Logic Teaching Tool

Solving logical exercises is usually done with pen and paper, but educational tools can offer useful pedagogical possibilities. The role of the educational software is to facilitate the student's grasp of the target procedures of education, and to provide teamwork and communication between teachers and students. Our teaching tool TABLEAUX (`http://www.fdi.ucm.es/profesor/rdelvado/FECS2011/`) is a prototype of an educational application based on propositional and first-order semantic tableaux with *equality* and *unification* [3] used as a support for the teaching of deductive reasoning at a very elementary university level for Computer Science and Software Engineering students. This tool helps our student to learn how to build semantic tableaux and to understand the philosophy of this proof device using it not only to establish consistency/inconsistency or to draw conclusions from a given set of premises but also for verification and debugging purposes.

2.1 Tool Usage

The tool consists of two main parts: one that produces tableaux and another based on the tableaux method for verification and debugging of algorithms. In both cases, the application possesses a drawing window where the trees will be graphically displayed (see Fig. 1). The user interacts with the provers through a graphical interface. We have chosen *Java*, but is possible to use *Prolog* to write the provers because its declarative character can give us a natural way to write the operations involved in the implementations [3].

2.2 Tool Implementation

An important design consideration in the tool implementation is that the code must be easy to maintain and extent, guaranteeing its future development and support in a sufficiently large portion of the Computer Science and Software Engineering education domain. We have made the choice of an open source *Java* code, facilitating the addition of new features for the verification and debugging of algorithms, and enabling changes to the tableau ruleset to accommodate these new methods and applications. This makes TABLEAUX more interesting for an educator to invest in the application and extension of this tool. Specific details on the straightforward aspects of a tableaux tool's development are described in [2], [5], [7]. We have selected the following aspects for a flexible and declarative representation of formulas and tableau rules:

- **Parsing and tokenizing formulas:** We have set up the tool in a declarative way defining all symbols that can be part of a well-formed formula in a symbol library and a graphical interface. The symbol library that is available to create formulas is declarative and extensible. The basic building block of the tool is the formula, represented internally as a parse that holds the formula's syntactic structure. By changing or extending the recursive definition and the symbol library, it is easy to expand the set of symbol strings accepted as well-formed to include First-Order logic and Floyd-Hoare logic.
- **Automatic tableau constructor:** The current implementation of the automatic prover built into the tool is straightforward and similar to other tableaux tools [2], [5], [7]. The automatic prover checks the rules applicable for a branch in the tableau, and selects the best one using a simple heuristic. Adapting the prover to give new alternative proofs for verification and declarative debugging is explained in the following sections.
- **Checking the tableau for mistakes:** An important functionality of the tool is to check the tableau for errors. Errors can occur in manually created tableaus in the three ways: Syntactic errors in resulting formulas, the wrong output for the correct rule, and applying the wrong rule. When checking the tableau for errors, ideally the application is able to discriminate between these possibilities. To check the tableau, the tool compares every applied rule with the correct and valid rules up to that point, to see if it constitutes a legal action and the results are correct.

3. Verification of Algorithms

The main novelty of the TABLEAUX tool is to train our students in the art and science of specifying correctness properties of algorithms and proving them correct. For this purpose, we use the classical approach developed by Dijkstra and others during the 1970s [1]. The tableau proof rules of the algorithm notation used in this paper provides new guidelines for the *verification of algorithms* from specifications (see [1] for more details). We use Dijkstra's guarded command language to denote our algorithms. Algorithms A are represented by functions `fun` A `ffun` that may contain variables (x, y, z, etc.), value expressions (e) and boolean expressions (B), and they are built out of the skip (`skip`) and assignment statements ($x := e$) using sequential composition (S_1`;`S_2), conditional branching (`if` B `then` S_1 `else` S_2 `fif`), and `while`-loops (`while` B `do` S `fwhile`). This language is quite modest but sufficiently rich to represent sequential algorithms in a succinct and elegant way.

It becomes obvious that neither tracing nor testing can guarantee the absence of errors in algorithms. To be sure of the correctness of an algorithm one has to prove that it meets its *specification* [1]. A specification of an algorithm A consists of the definition of a *state* space (a set of program variables), a *precondition* P and a *postcondition* Q (both predicates expressing properties of the values of variables), denoted as $\{P\}\ A\ \{Q\}$. Such a triple means that Q holds in any state reached by executing A from an initial state in which P holds. An algorithm together with its specification is viewed as a theorem. The theorem expresses that the program satisfies the specification. Hence, all algorithms require proofs (as theorems do). Our tool verifies algorithms according to their specification in a constructive way based on semantic tableaux $P \vdash \neg\, wp(A, Q)$, where $wp(A, Q)$ is the *weakest precondition* of A with respect to Q, which is the 'weakest' predicate that ensures that if a state satisfies it then after executing A the predicate Q holds (see [4] for more details).

As an illustrative example, we consider the formal verification of a simple algorithm $divide$ to compute the positive integer (int) division between a and b (with quotient c and remainder r), specified as:

$\{P\ :\ a \geq 0\ \wedge\ b > 0\}$
`fun` *divide* $(a, b\ :\ int)$ `dev` $< c, r\ :\ int >$
  $c := 0;\quad r := a;$
  $\{I\ :\ a = b * c + r\ \wedge\ r \geq 0\ \wedge\ b > 0\ ,\ C : r\}$
  `while` $r \geq b$ `do`
    $c := c + 1;\quad r := r - b$
  `fwhile`
`ffun`
$\{Q\ :\ a = b * c + r\ \wedge\ r \geq 0\ \wedge\ r < b\}$

Following [4], the verification is based on a *loop invariant* I (supplied by the designer or by some invariant-finding tool), a *bound function* C (for termination), and five proofs:

- $\{P\}\ c := 0\,;\ r := a\ \{I\}$.
- $\{I\ \wedge\ r \geq b\}\ c := c + 1;\ r := r - b\ \{I\}$.
- $I\ \wedge\ r < b \Rightarrow Q$.
- $I\ \wedge\ r \geq b \Rightarrow C \geq 0$.
- $\{I \wedge r \geq b \wedge C = T\}\ c := c{+}1;\ r := r{-}b\ \{C < T\}$.

Our tool represents each of these proofs as a *closed* semantic tableau. We assume the reader is familiar with the classical tableau-building rules (α and β), equality ($=$), and closure rules (see [3] for more explanations and details). We use the notation $R^{e,\cdots}_{x,\cdots}$ to represent the predicate R in which x is replaced by e, etc. For example, we have the following tableau proof (graphically displayed by the TABLEAUX tool in Fig. 2) to verify the preservation of the invariant I in the body of the loop: $\{I\ \wedge\ r \geq b\}\ c := c + 1;\ r := r - b\ \{I\}$ $\Leftrightarrow I \wedge r \geq b \vdash \neg\, wp\,(c := c{+}1\,;\ r := r - b, I) \Leftrightarrow I\ \wedge\ r \geq b$ $\vdash \neg\,(I\ ^{c+1,\,r-b}_{c,\,r})$.

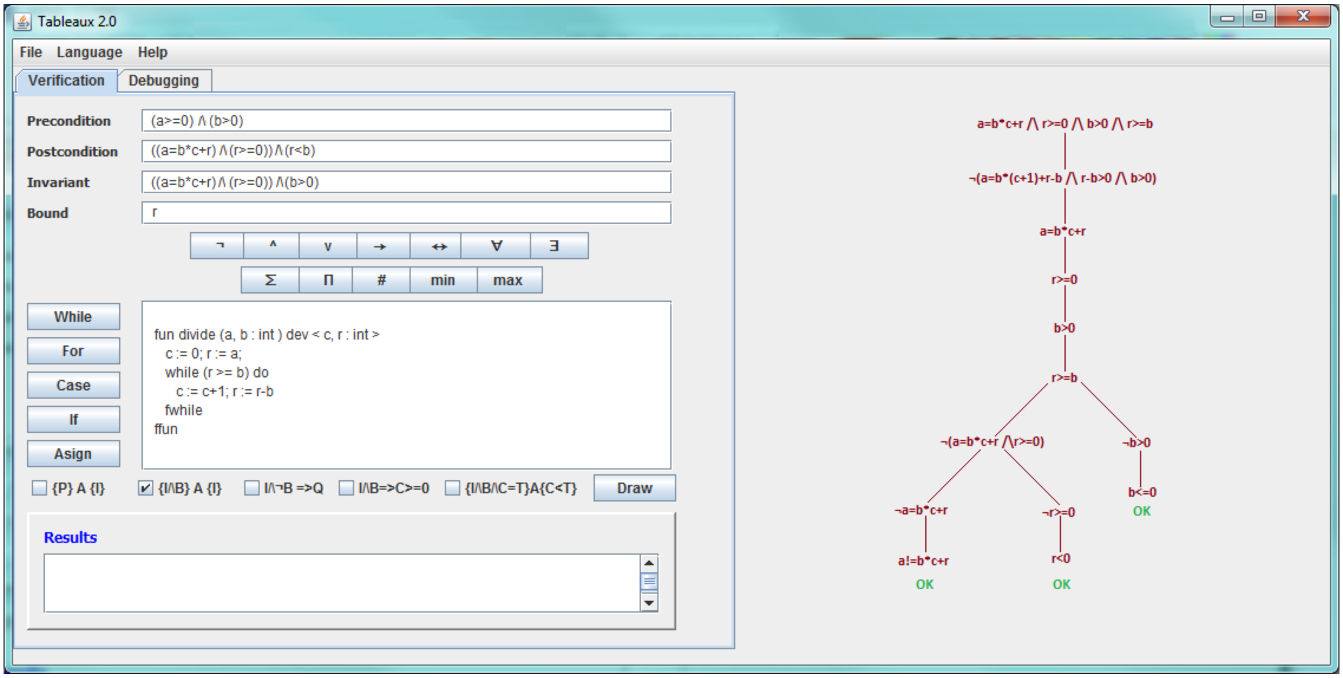


Fig. 2: The logic teaching tool TABLEAUX for the formal verification of algorithms by mean of semantic tableaux.

(1) $a = b*c+r \wedge r \geq 0 \wedge b > 0 \wedge r \geq b$		$\{I \wedge r \geq b\}$
(2) $a = b*c+r$		$(\alpha, 1)$
(3) $r \geq 0$		$(\alpha, 1)$
(4) $b > 0$		$(\alpha, 1)$
(5) $r \geq b$		$(\alpha, 1)$
(6) $\neg(a = b*(c+1)+r-b \wedge r-b \geq 0 \wedge b > 0)$		$\{\neg(I^{c+1,\, r-b}_{c,\, r})\}$
		$(\beta, 6)$
(7) $a \neq b*c+r$	(8) $r < b$	(9) $b \leq 0$
$\times\ (2,7)$	$\times\ (5,8)$	$\times\ (4,9)$

Finding invariants is an essential part of this educational process. We can use the tool to guide our students to obtain loop invariants from specifications. For example, if we only provide to our students the postcondition Q, they usually infer only an incomplete predicate I' : $a = b*c+r$ as the loop invariant. Then, when they apply the tool to verify the algorithm, they obtain an *open* semantic tableau (indicated by $\times$) for $I' \wedge r < b \Rightarrow Q$:

(1) $a = b*c+r \wedge r < b$		$\{I' \wedge r < b\}$
(2) $a = b*c+r$		$(\alpha, 1)$
(3) $r < b$		$(\alpha, 1)$
(4) $\neg(a = b*c+r \wedge r \geq 0 \wedge r < b)$		$\{\neg Q\}$
		$(\beta, 4)$
(5) $a \neq b*c+r$	(6) $r < 0$	(7) $r \geq b$
	$\Downarrow$	
$\times\ (2,5)$	$\times$	$\times\ (3,7)$
	$\Downarrow$	

We need to insert $\boxed{r \geq 0}$ in I' to close this tableau

From the open branch, our students learn to complete the invariant with the predicate I'' : $a = b*c+r \wedge r \geq 0$. However, they still have an open tableau for $\{I'' \wedge r \geq b \wedge C = T\}$ $c := c+1;\ r := r-b$ $\{C < T\}$:

(1) $a = b*c+r \wedge r \geq 0 \wedge r \geq b \wedge r = T$	$\{I'' \wedge r \geq b \wedge C = T\}$
(2) $a = b*c+r$	$(\alpha, 1)$
(3) $r \geq 0$	$(\alpha, 1)$
(4) $r \geq b$	$(\alpha, 1)$
(5) $r = T$	$(\alpha, 1)$
(6) $r - b \geq T$	$\{\neg(C < T)^{c+1,\, r-b}_{c,\, r}\}$
(7) $b \leq 0$	$(=, 5, 6)$
$\Downarrow$	
$\times$	

$\Rightarrow$ We need to insert $\boxed{b > 0}$ in I'' to close this tableau

Finally, they learn to insert $b > 0$ in the assertion I'' to complete the loop invariant I. If they apply the tool again, all the tableaux remain closed and the formal verification session finishes.

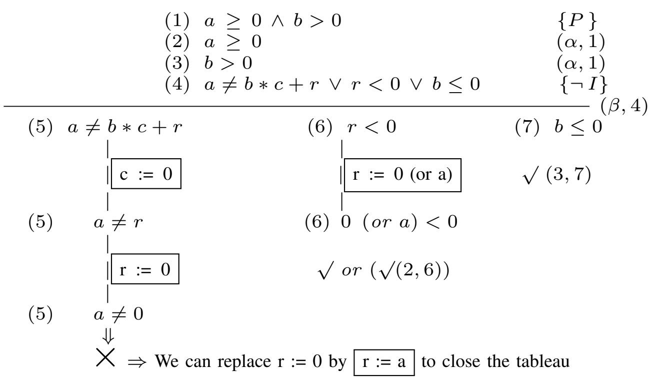
4. Algorithmic Debugging

Debugging is one of the essentials parts of the software development cycle and a practical need for helping our students to understand why their programs do not work as intended. In this section we apply the ideas of *algorithmic debugging* [6] as an alternative to conventional approaches to debugging for imperative programs. The major advantage of algorithmic debugging compared to conventional debugging is that allows our students to work on a higher level of abstraction. In particular, we have successfully applied our tool based on semantic tableaux for the algorithmic debugging of simple programs to show how one can reason about such programs without operational arguments. Moreover, the methodology provides an excellent training in the calculus needed for *program derivation* [4].

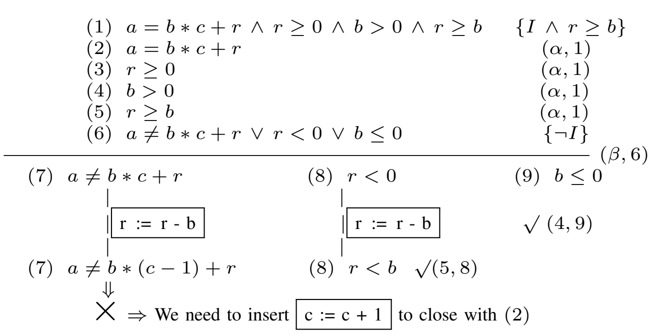
Following a seminal idea from Shapiro [8], algorithmic debugging proposes to replace computation traces by *computation trees* with program fragments attached to the nodes. As novelty, in this work we propose to use computation trees as semantic tableaux. As an example, we alter the code of the previous algorithm for integer division with two mistakes:

```
{P : a ≥ 0 ∧ b > 0}
fun divide (a, b : int) dev < c, r : int >
  c := 0; [r := 0];                 ← wrong code!
  {I : a = b*c + r ∧ r ≥ 0 ∧ b > 0 , C : r}
  while r ≥ b do
              [r := r - b]          ← missing code!
  fwhile
ffun
{Q : a = b*c + r ∧ r ≥ 0 ∧ r < b}
```

If we try to verify this erroneous algorithm, we can execute again the tool. Now, TABLEAUX displays an open semantic tableau $P \vdash \neg I$ for debugging $\{P\}\, c := 0; r := 0\, \{I\}$, instead of $P \vdash \neg(I^{0,0}_{c,r})$. However, the weakest precondition $I^{0,0}_{c,r}$ is built from (5) and (6), step by step, to identify erroneous parts of the code in open branches (see Fig. 4):

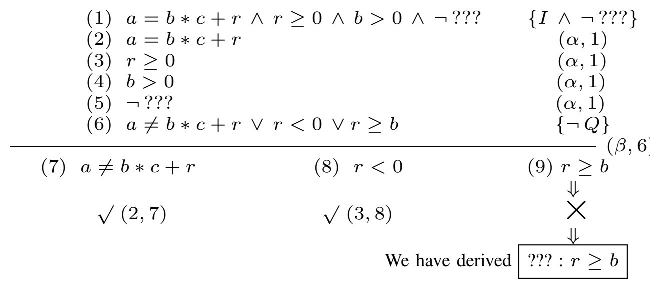


After this correction, we obtain a closed tableau. Now, we can execute again the tool to perform the algorithmic debugging of $\{I \wedge r \geq b\}\ r := r - b\ \{I\}$:

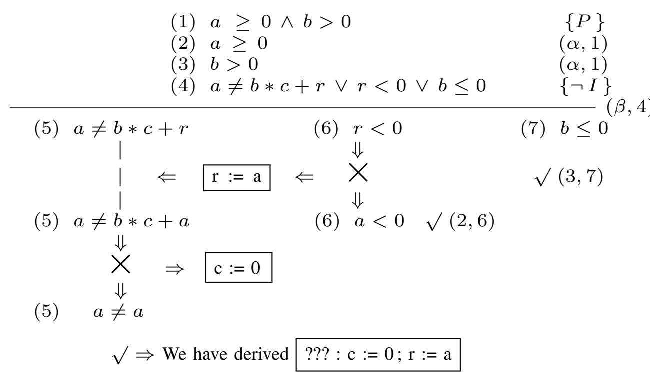


To close the open branch, we infer that we need to insert new code. This particular incompleteness symptom could be mended by placing $c := c + 1$ in the body of the loop. If we apply again the tool, no more errors can be found and the five tableaux remain closed. The debugging session has finished.

This algorithmic debugging methodology can be also applied to explain the *derivation of simple algorithms* [4]. For example, the semantic tableau for $I \wedge \neg(???) \Rightarrow Q$ allows to our students to derive the repetition condition:



Analogously, we can display a semantic tableau for $\{P\}$??? $\{I\}$ to derive the initialization code:



Along the academic course 2009/2010, we have applied this programming technique to design more specific and illustrative classes of problems [4]. We believe that all these examples with tableaux provide an excellent training in the calculus and reasoning needed for deriving correct programs.

5. Experiences and Results

The prototype of the logic tool is available for the students of the topics *Computational Logic* and *Design of Algorithms* in the Computer Science and Software Engineering Faculty of our University through a Virtual Campus `http://www.ucm.es/campusvirtual/`. The following results are based on the statistics from the 486 students who took the course in 2009/2010.

5.1 Design of the Experiences

We have carried out two educational experiences:

	All the questions						Excluding algorithm analysis questions					
	Correct		errors		don't knows		Correct		errors		don't knows	
	media	σ	media	σ	media	σ	media	σ	media	σ	media	σ
slides/books	9.36	2.73	6.2	2.3	3.2	2.8	9.18	2.2	4.31	1.8	0.57	1.58
tableaux/tool	11.8	2.87	4.8	2.1	1.2	1.7	9.51	2.5	4.2	2.1	0.66	1.55

Fig. 3: Medias and standard deviations (σ) of the controlled experience.

- One **non-controlled** experience: All the students may access the Virtual Campus and participate freely in the experience: download and use the tool, and answer different kinds of tests.
- One **controlled** experience: Two groups of students must answer a test limited in time and access to material.

With respect to the **non-controlled** experience, the students may freely access the Virtual Campus without any restriction of time (but the last delivery time at the end of the course) or material (slides, tool, bibliography) and answer the questions of several tests.

For each of the following topics in Computer Science and Software Engineering we have provided a test that evaluates the knowledge of our students applying different kinds of semantic tableaux. The students may use these tests to verify their understanding of the different concepts. The questions are structured in three blocks: *propositional and predicate logic*, *specification and verification* of algorithms, and *debugging and derivation* of imperative programs. The resolution of the tests by the students is controlled by the Virtual Campus with the help of an interactive tutoring system.

In the **controlled** experience we try to evaluate more objectively the usefulness of the tool. In particular we have chosen the application of TABLEAUX for the verification and debugging of simple searching and sorting algorithms, and segment problems [4].

5.2 Obtained Results

5.2.1 Non-controlled experience

We outline here the main conclusions from the results of the non-controlled experience. With respect to the material the students used to study, as long as the contents are more complicated the use of the tool (simulation, case execution, and tool help) increases considerably. Better results are obtained in the verification and debugging of searching and sorting problems (linear and binary search, insertion and selection sort). The tool helps to visualize array manipulations in array assignments. In the rest of the algorithms (segment problems, slope search, and advanced sorting algorithms) they used only class material or bibliography.

When answering the tests questions, the students were also asked whether they needed additional help to answer them. In the case of linear and binary search they use the tool as much as the class material, which means that visualization of their own trees are useful for them. In the case of sorting algorithms, they used "other" material a bit more than the tool, which presumably consists of a colleague advice.

We can conclude that although only a small percentage of students have followed this experience, those who have done it, both consider the tool as an interesting material and have used it to complement the rest of the available material, including the class explanations. The tool is specially useful for visualizing and mechanize many of the boring and routine aspects of verification and algorithmic debugging.

5.2.2 Controlled experience

This experience was carried out with 59 students. We gave 32 of them the slides of the course and the books [3], [4]. The rest were taken to a computer laboratory, where they could execute the TABLEAUX tool. We gave the same test to both groups, consisting of 18 questions, 12 of them about specification aspects of the algorithms (infer invariants and bound functions), and the rest about their verification and debugging.

In Fig. 3 (left) we provide the media and the standard deviation of the correct, the errors, and the *don't knows* answers. First, we observe that students using the TABLEAUX tool answer in media more questions than the other ones. In addition, they make less errors than the others. Fig. 3 (right) gives the results excluding questions about the analysis of algorithms. Now both groups present a similar behavior. The standard deviation of the students belonging to the *slides/books* is slightly lower than that of the other group. The explanation for these results may be that students who use the tool assimilated the concepts in a more homogeneous way, while the use of more conventional learning material, such as books or slides, leads to greater differences among the students.

Finally, we can conclude that the methodology proposed in this work constitutes an optimum complement to facilitate the comprehension of the analysis of different algorithms. In addition, it has helped us to detect in the students difficulties to apply the programming techniques to derive algorithms.

The most positive aspect of this experience is that all the

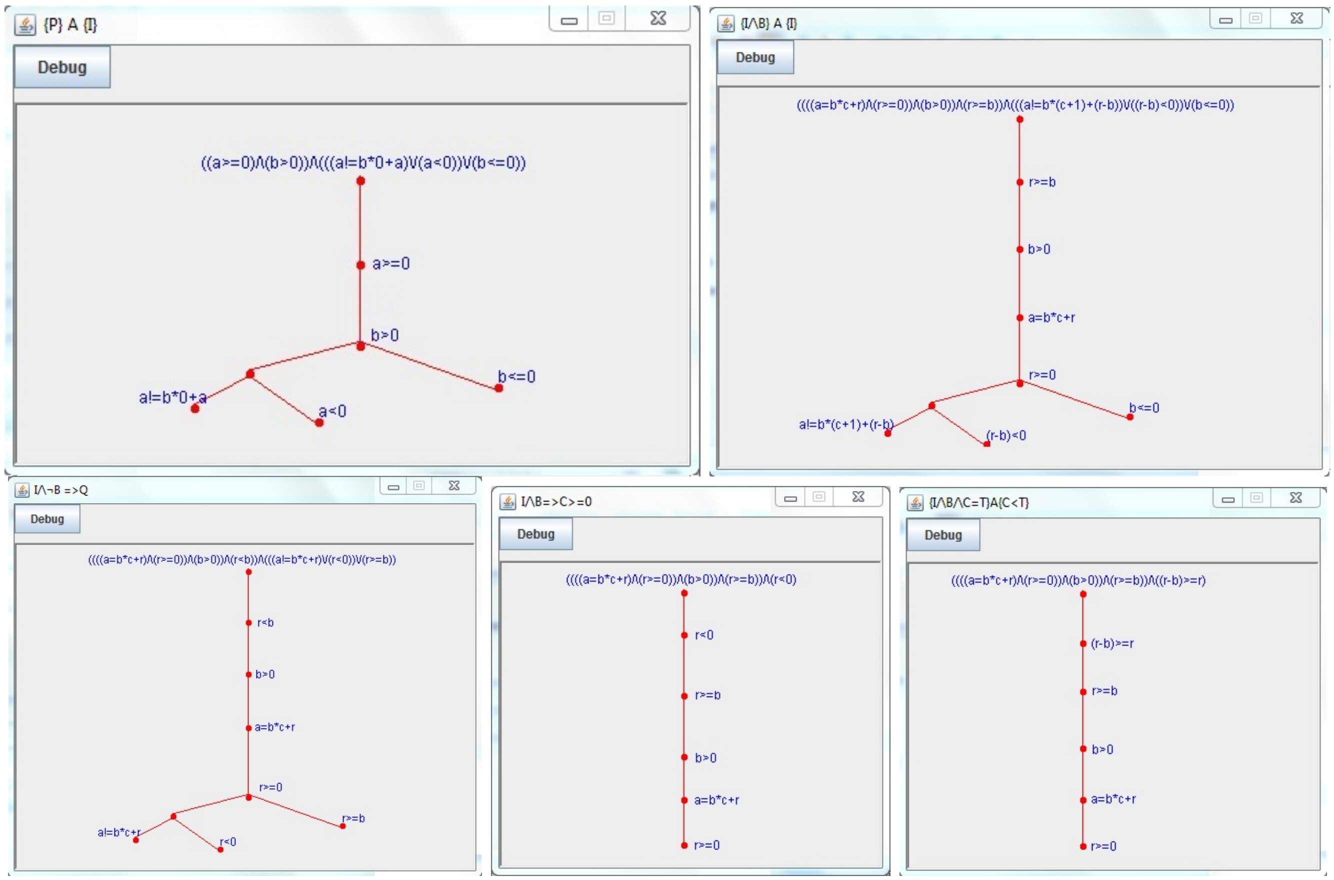


Fig. 4: The logic teaching tool `TABLEAUX` for the algorithmic debugging of programs by mean of semantic tableaux.

students who used the TABLEAUX tool indicated the benefits of using them to understand the design and analysis of the algorithms. They also mentioned that consulting written material would also be helpful to answer the questions about the algorithms analysis. Some of them also complained about examples of algorithms without specification or invariants, as it were not as representative of all cases of the algorithm as it were desirable. They argued that the examples must exhibit the complete behavior of the algorithm, avoiding possible misunderstandings of it. However, we believe that our students may be guided by the tool to make right assumptions about the formal properties of the algorithms, as we have seen in Sections 3 and 4.

6. Conclusions

We have presented a software engineering educational tool based on semantic tableaux for a specification language on predicate logic. This is the first step towards the development of a practical reasoning tool for formal verification and declarative debugging of algorithms.

We have systematically evaluated the proposed method to confirm that a tableaux tool is a good complement to both the class explanations and material, making easier the visualization of proofs in the reasoning needed for the design of correct and efficient software.

Acknowledgements

This work has been partially supported by the Spanish project PIMCD 2010/97 (Project for the Innovation and Improvement of the Educational Quality).

References

[1] E. Dijkstra. *A Discipline of Programming*. Prentice Hall, 1976.
[2] H. van ditmarsch. *Logic software and logic education*. Contains a comprehensive list of educational logic software, 2005.
[3] M. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer, Graduate Texts in Computer Science, 1990.
[4] A. Kaldewaij. *Programming: The Derivation of Algorithms*. Prentice-Hall International Series in Computer Science, 1990.
[5] B.P. Lancho, E. Jorge, A. de la Viuda, and R. Sanchez. *Software Tools in Logic Education: Some Examples*. Logic Journal of the IGPL 15(4): 347-357, 2007.
[6] L. Naish. *A Declarative Debugging Scheme*. Journal of Functional and Logic Programming, vol. 3, 1997.
[7] R. de Queiroz. *Archive logic-l*. Logic programs and teaching aids, theorem provers and languages.
[8] E.Y. Shapiro. *Algorithmic Program Debugging*. MIT Press, Cambridge, MA, USA, 1983.

Implementation of a Web-based Virtual Laboratory for Electronic and Digital Circuit Experiments Using Java Applets

Dongsik KIM, Jaeho CHA, Youngji KIM, Changwan JEON, Kwansun CHOI, and Jongsik LIM

Department of Electrical and Communication Engineering, Soonchunhyang University, KOREA

dongsik@sch.ac.kr

Abstract – *This paper presents web-based virtual laboratory system for electronic and digital circuit experiments. Through our virtual laboratory, the learners will be capable of learning the concepts and theories related to circuit experiments and how to operate the virtual experimental equipments such as multimeters, function generators, digital oscilloscopes, DC power suppliers and bread board etc.*

The proposed virtual laboratory system is composed of five important components: Principle Classroom to explain the concepts and theories of electronic and digital circuit operations, Simulation Classroom to provide a web-based simulator to the learners, Virtual Experiment Classroom to provide interactive Java applets about the syllabus of off-line laboratory class, Assessment Classroom, and Management System. With the aid of the Management System every classroom is organically tied together collaborating to achieve maximum learning efficiency. We have obtained several affirmative effects such as high learning standard, reducing the total experimental hours and the damage rate for experimental equipments.

Keywords: Web-based Virtual Laboratory, Digital Circuits, Electronic Circuits, Conceptual Java Applets,

1. INTRODUCTION

Recently, much interest has been drawn on the web-based solution for the experiments at universities with large number of students due to low cost. This interest is mainly due to the cost of the experimental laboratories at universities with a large number of students. The world-wide web provides new opportunities for distributing all learning materials over the internet. The worldwide web, when combined with other network tools, can be used to provide useful educational information to learners.

But, the variance of students' learning capability also make difficult for the educators to maintain the quality of educational services. Also, the simple and tedious webpage which is composed of text and image-based learning materials has made the lowering of educational efficiency in the worldwide web. The predominantly one-way nature of the media used in planning and preparation of learning materials results in little or no student-to-student and student-to-teacher communication. This lack of interaction leads to students' feeling isolated.

To cope with these difficulties, web-based virtual laboratory system provides new opportunity for solving these problems with paying reasonable cost [1-3]. Thus, this paper presents a virtual laboratory system which can easily be used on the web by simple mouse manipulations. The proposed virtual laboratory system provides improved learning methods which can enhance the multimedia capabilities of world-wide web. If the learners have access to the virtual laboratory system through a typical web browser such as Internet Explorer, they can make experiment on basic electrical circuits through simple mouse clicks. Since this interactive virtual laboratory is implemented to describe the actual on-campus laboratory, the learners can obtain similar experimental data through it.

The proposed virtual laboratory system is composed of five important components: *Principle Classroom, Simulation Classroom, Virtual Experiment Classroom, Assessment Classroom and Management System*. Our system supports from elementary electronic and digital experiments to advanced electronic experiments included in the curriculum of the college of engineering. It has interactive and innovative multimedia contents to get the learners exact understanding of the concepts and theories of circuit operations, and the learners can build and simulate their own circuits and measure all information about the status of the circuits on the virtual space by simple mouse manipulation. Every activity done in the virtual laboratory is recorded on database and provided to the learners as a printout form including experimental information and results. The educators check the submitted printout form to estimate how well the learners understand the experimental contents. Our system provides 4 courses and each course needs one semester (16 weeks).

If the learners have access to the virtual laboratory system through a typical web browser such as Internet Explorer, they can make an experiment through simple mouse clicks. The implemented virtual laboratory system can be used in stand-alone fashion, but using as assistants of the actual on-campus laboratory class shows more encouraging results.

2. COURSEWARE STRUCTURES

We have 4 virtual laboratory courses for undergraduate. The material in first and second courses is appropriate for elementary courses on electronic and digital circuit experiments and the material in third and fourth courses for advanced courses on electronic circuit experiments. Each course consists of 16 chapters and each chapter comprises *the Principle Classroom* to explain the concepts and theories of circuit operations, *the Simulation Classroom* to provide a web-based simulator to the learners, *the Virtual Experiment Classroom* which provides interactive and innovative multimedia contents to build and test several circuits. *The Management System* gives the learners and the educators their ID and password and provides printout services for all information about experiment done *in the Virtual Experiment Classroom* [3-7]. In Fig. 1, the structure diagram of our virtual laboratory system is shown.

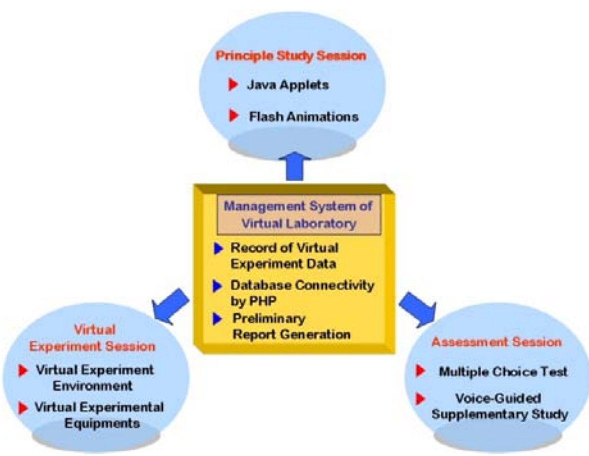


Fig. 1 Structure of our Virtual Laboratory System

2.1 Principle Classroom

The Principal Classroom is responsible for making the learners understand the concepts and theories of the circuit operations included in each chapter. Interactive Java applets with creative and intuitive ideas for each subject lead the learners to easily understand their operations. On-line voice presentation and its related texts together with moving images are synchronized for efficient learning process.

Figure 2 shows several important procedures from the java applets for explaining the concepts of 2 to 4 decoder. The conceptual Java applet in Fig. 1 is authored to let the learners easily understand the principle of decoder by clicking the several buttons such as “Move”, “One click”, “Show grid”, “Detail”, “Reset” and “Measure”.

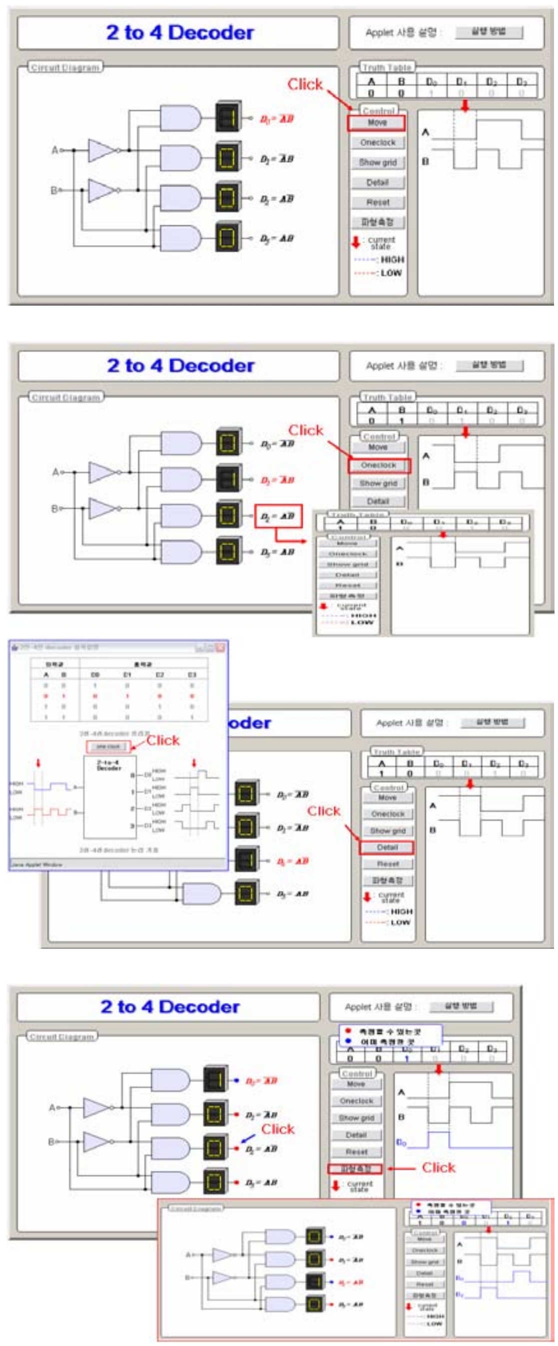


Fig. 2 A conceptual Java applet for 2 to 4 decoder

Fig. 3 shows a Java applet for understanding the concepts of collector characteristic curves. Using a transistor circuit in Fig. 3, the learners can generate a set of collector characteristic curves that show how the collector current varies with the collector-to-emitter voltage for specified value of base current.

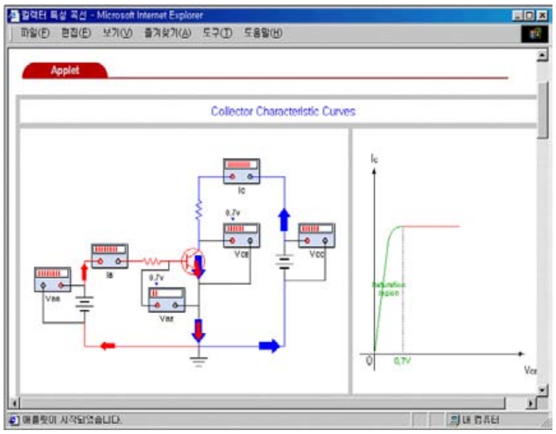
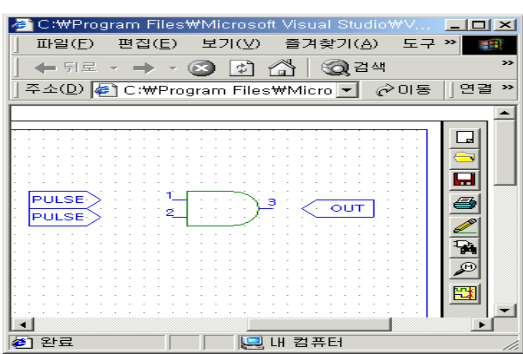


Fig.3. A Java applet for collector characteristics

2.2 Simulation Classroom

The Simulation Classroom provides a web-based digital simulator to the learners, from which they can simulate several digital circuits for various input conditions. The proposed digital simulator is implemented to have several simplified functions which are essential to the learning process of digital logic circuits. The learners by themselves simulate several digital logic circuits on the web for specific input conditions and the design/analysis of digital logic circuits can be available. Furthermore, two or more different digital circuits can be simulated simultaneously for different input conditions. The proposed simulator, combined with multimedia contents, can be used as an auxiliary educational tool and can enhance the improved learning efficiency.



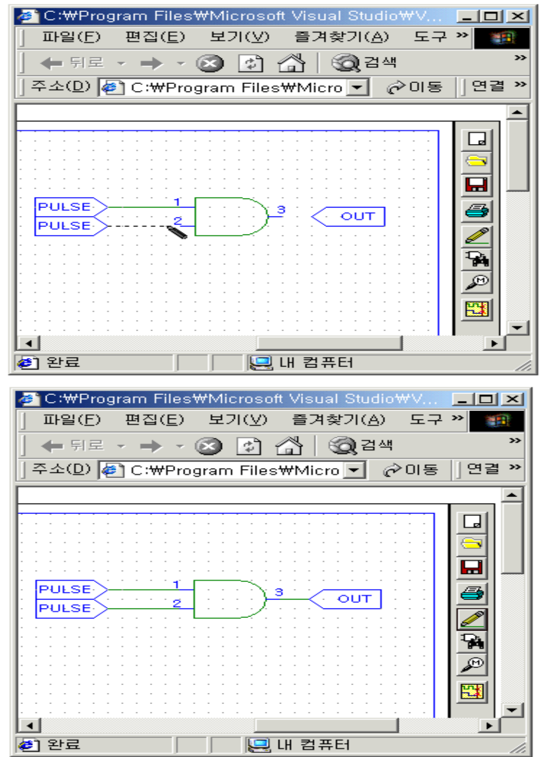


Fig. 4 A digital simulation procedure for logic gates

Fig. 5 shows a web-based digital simulator which is performing simulation for logic gates. The simulation shown in Fig. 5 is performed according to the following procedure: (1) Circuit Composition on the Layout Grid (2) Applying Input Voltage (3) Output Measurements.

2.3 Virtual Experiment Classroom

The Virtual Experiment Classroom provides virtual experimental environment to the learners. Widely used experimental equipments such as oscilloscopes, multimeters, function generators and power supply etc. are implemented by Java Applets. In this classroom, the learners can build circuits for each subject, set the values for each circuit element, and measure voltages or currents etc. using the experimental equipments. When finishing the virtual experiment on the web, the learners can print out the all information related to the experiment which can be used as preliminary report for on-campus laboratory class

For example, The virtual experiment is performed according to the following procedure: (1) Circuit Composition (2) Applying Input Voltage (3) Output Measurements (4) Transmission of Experimental Data (5) Printout of Preliminary Report as shown in Fig. 5 (a)-(d). The learners build a given circuit by placing proper circuit elements from ELEMENT CHOICE menu. In this menu, the learner can select circuit elements and change their types or values. The learners can change the value of DC power supply by double-clicking the DC power supply symbol. Also, the learners can insert a voltage and/or current markers into the circuit by using MEASURE menu. The learner can also measure several outputs for the various values of V_{DD} using the oscilloscope. In Fig. 6, overall virtual experiment for common-source (CS) JFET amplifier is illustrated to show the experiment procedure.

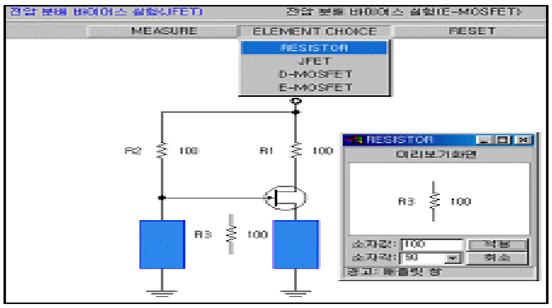


Fig. 5(a) Circuit Composition

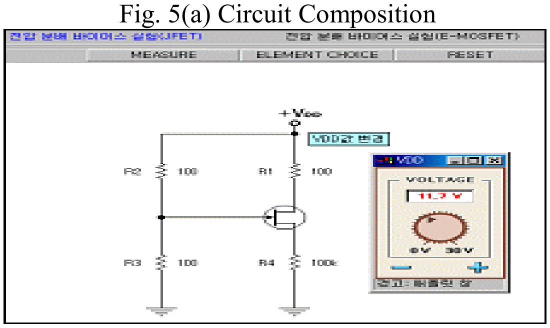


Fig.5(b) Applying Input Voltage

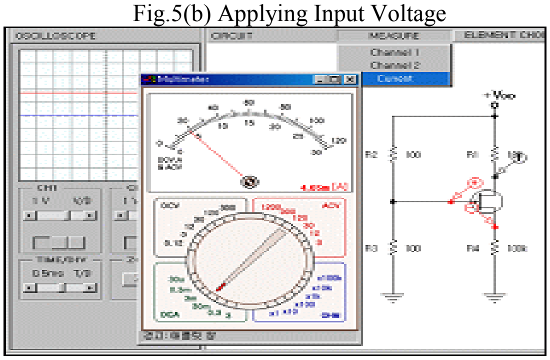


Fig. 5(c) Output Measurements

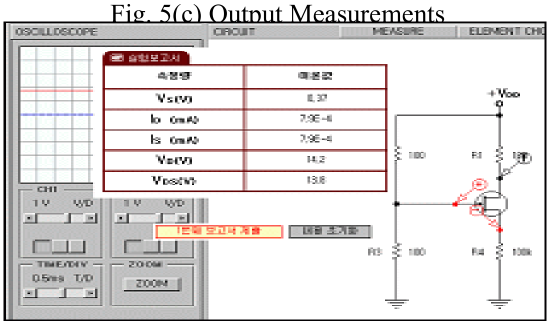


Fig. 5(d) Transmission of Data

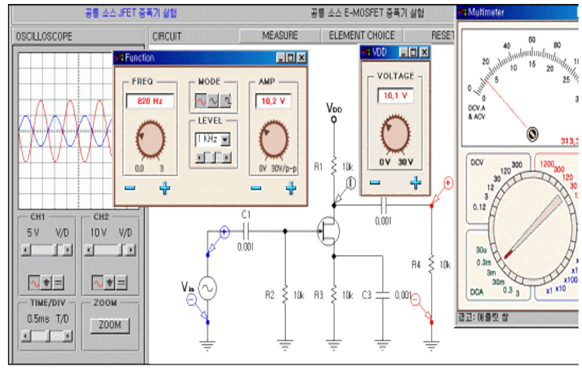


Fig. 6 Virtual experiment for CS JFET amplifier

The VLab system is designed and implemented in the form of Java applets by introducing the following step-by-step procedure as shown in Fig. 7: (1) Design and display basic units of the VLab system for digital logic gates. (2) Handle mouse events by appropriate event handlers. (3) Display the circuit connection on the VBB and its corresponding online schematic diagram. (4) Measure and display the outputs of the circuits to be performed on the VBB of the VLab system. (5) Design and implement the data transmission to the database.

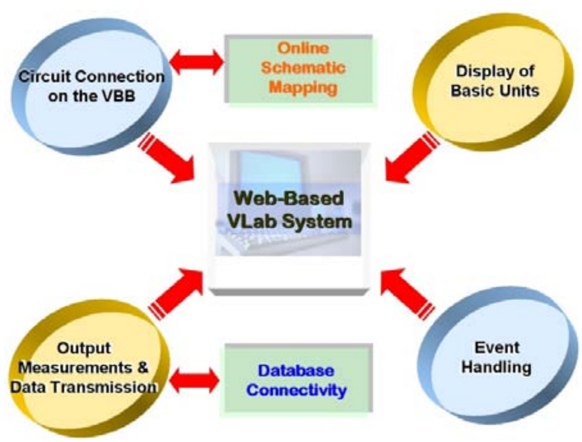


Fig. 7 Design procedure of the VLab system

The Virtual Experiment Classroom also has an efficient virtual experiment kit with interactive and innovative multimedia contents, which can be used to enhance the quality of education in the area of digital logic circuits. A Java applet for virtual experiment on decoder/encoder is illustrated in Fig. 8(a). In Fig. 8(b) Java applet for virtual experiment on JK-flip flop is illustrated. Note that the circuit composition on the virtual bread board and its corresponding online schematic diagram are displayed together on the virtual experiment kit for the learner's convenience.

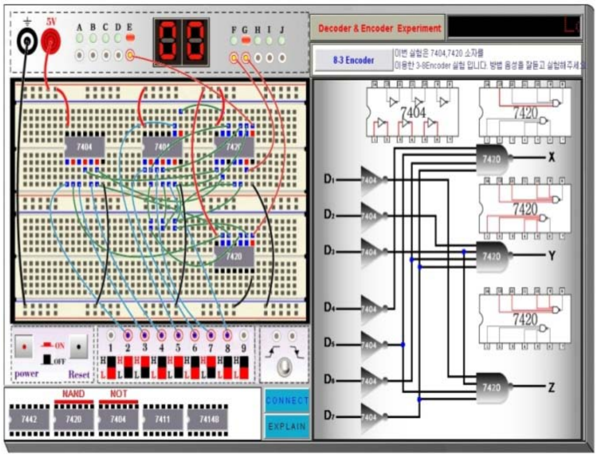


Fig. 8(a) Virtual experiment for decoder/encoder

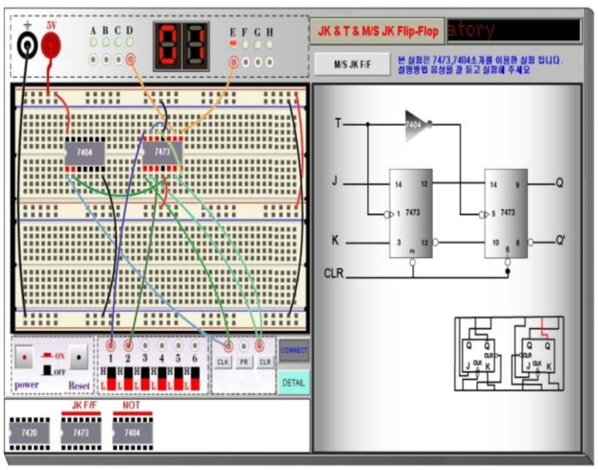


Fig.8(b) Virtual experiment for JK flip flop

Fig. 9 shows an applet for virtual experiment on decoder/encoder. This applet shows the process of measuring the outputs under various input conditions.

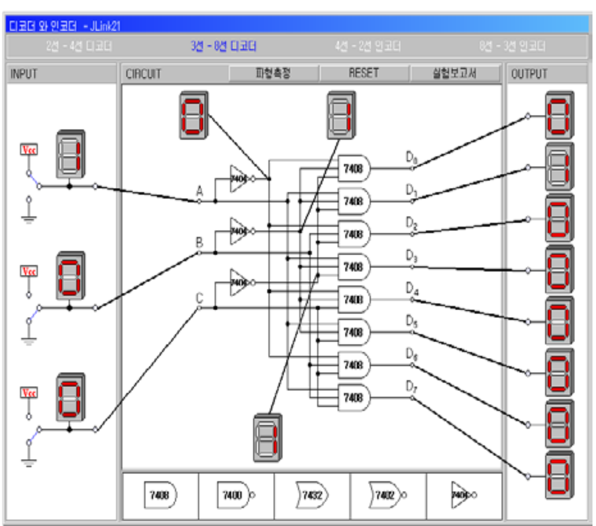


Fig. 9 Java applet for decoder/encoder

2.4 Assessment Classroom

It is very important to provide the educators with useful information about experiments done in virtual laboratory by which the educators evaluate how well the learners are doing. Every activity done in the virtual laboratory is recorded on the database and provided to the learners as a printout form including experimental information and results. The educators check out the submitted printout form to estimate how well the learners understand the experimental procedures. *The Management System* supports communications between the educators and the learners in the ways mentioned above, and different setups for each learner. Our system based on the client/server architecture uses noncommercial software. Furthermore, simple multiple choices are given to the learners after virtual experiments and the test results are displayed on the message box. According to the test result for each question, if the learners click one of two buttons named as "supplementary study" or "more challenging study", the learners can listen to the voice regarding the related explanations. This assessment process is very essential to increase the learner's academic capability.

2.5 Management System

Good instructional development is an iterative process by which the educators and learners perform formative assessments and summative evaluations to continually improve a course. Effective instructors use a variety of means, some formal and others informal, to determine how much and how well their students are learning. In the proposed virtual laboratory system, every activity occurred in the virtual laboratory will be recorded on database and printed out on the preliminary report form. All of this can be achieved by the aid of Management System. The database connectivity is made by Professional HTML Preprocessor and the virtual laboratory environment is set up slightly differently for each learner. Our virtual laboratory system, based on client/server architecture, uses none of the commercial software package. Fig. 10 shows database connectivity of *the Management System* using PHP. Also, Fig. 11 shows a preliminary report form to be printed out after virtual experiment.

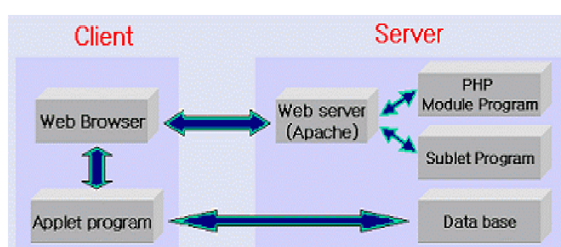


Fig. 10 Database connectivity of virtual laboratory

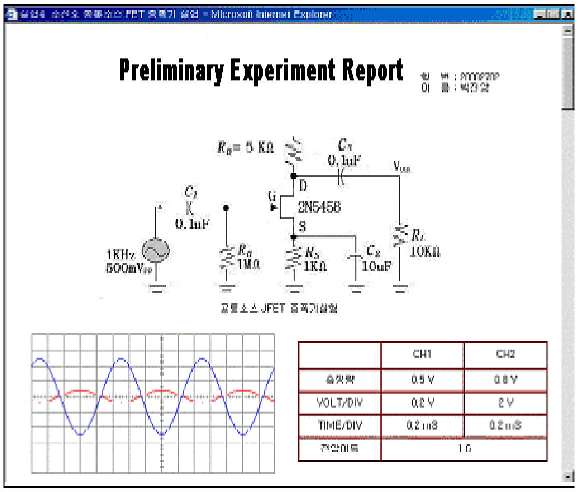


Fig. 11 A preliminary report form

3. CONCLUSIONS

We implement an efficient virtual laboratory system with creative and interactive multimedia contents, which can be used to enhance the quality of education in the area of electronic and digital circuit experiments.

Our system shows that the difficult concepts, principles and theories related to the experiments can be conveyed to the learners effectively by interactive and innovative multimedia contents. The implemented virtual experimental equipments such as oscilloscopes, multimeters and function generators can be good examples of educational auxiliary tools.

The system has brought several affirmative effects such as reducing the waste time and labor of both the educators and the learners, and the damage rate of real equipments, and increasing learning efficiency as well as faculty productivity. The implemented virtual laboratory system can be used in stand-alone fashion, but using as assistants of the actual on-campus laboratory class is recommended.

REFERENCES

[1] Dongsik Kim et al," Practical Implementation of A Web-based Virtual Laboratory in the Area of Electrical Engineering," IASTED International Conf. on Computers Advanced Technology in Education, 2001.

[2] Dongsik Kim et al, "Implementation of a Web-based Virtual Laboratory System for Electronic Circuit Experiments," International Conference on Internet Computing, Las Vegas, Nevada, USA, 2003.

[3] Kwansun Choi, Dongsik Kim and Sunheum Lee, "A Study on the Development of a Web-based C Compiler for C Programming Practice," International Conf. on Internet Computing, Las Vegas, Nevada, USA, 2003.

[4] K. Kim, C.J. Bonk, T. Zeng, "Surveying the future of workplace e-learning: the rise of blending, interactivity, and authentic learning", http:// www.elearnmag.org, 2005.

[5] L. Sun, Y. Fu, "Interoperability for eLearning Services Management and Provision, Journal of World Wide Web: Internet and Web Information System, Vol. 8, pp1-18, 2005

[6] J. Lubega, "A Design for Generating Personalized Feedback in Knowledge Construction, pp 302-310, Springer, Heidelberg, 2005.

[7] J. Lubega, " A Study of Factors Contributing to Late Submission of Course Work," European Journal of Open, Distance, and E-learning (EURODL), Vol.2, 2006.

[8] T.A. Fjeldly, M.S. Shur,"Lab on the Web: Running Real Electronics Experiments via the Internet,"Wiley-VCH, Chichester, 2003

[9] M. Khalifa, R. Lam," Web-based Learning: Effect on Learning Process and Outcome," IEEE Transactions on Education, p.350-356, 2002.

[10] D. Consonni, C. Seabra," A Modern Approach to Teaching Basic Experimental Electricity and Electronics," IEEE Transactions on Education, p. 5-15, 2001.

Construction of a Collaborative Learning Environment through Sharing of a Single Desktop Screen

Masanori NAKAKUNI, Masaru OKUMURA and Sho FUJIMURA

Information Technology Center, Fukuoka University
19-1, Nakanuma 8, Jounan-ku, Fukuoka 814-0180 JAPAN
nak@fukuoka-u.ac.jp, okkun@fukuoka-u.ac.jp, fujimura@relativity.rd.fukuoka-u.ac.jp

Abstract - *In this study, we constructed an environment conducive to enhancing education efficacy through an improvement of collaborative learning in higher education. In the process of collaborative learning, one faces a variety of situations, for instance, "discussion on learning themes" and "work to draw up materials and reports pertaining to learning." One of the styles of collaborative learning is to proceed with discussion or work by exchanging opinions among participants while writing down figures and words on a single blackboard. In this manuscript, we propose to replace the blackboard with a tablet PC with the view of omitting complicated tasks involved in each process of learning to facilitate its progress, thereby creating a collaborative learning environment where the participants can concentrate on learning itself. Furthermore, this manuscript also refers to the results of a collaborative learning program that was actually implemented on computer programming. Lastly, we touch on some advantages and issues associated with such learning environments including application in cases where the participants of a learning program live separately in remote areas.*

Keywords: Group Learning, Communication, Collaboration, Support, Education

1 Introduction

In a typical collaborative learning program, learning activities are performed on a group basis, with each group consisting of several members. This is a form of learning in which members within a group cooperate with each other in order to achieve a shared goal. By undergoing a collaborative learning program of this kind, a participant can expect to gain educational effects more easily than by learning by him/herself. A collaborative learning program involves the participants influencing with one another in the process of proceeding with learning, such as sharing knowledge and exchanging opinions with others, as well as expressing themselves. It is certainly true that shortening the time required for resolving problems through sharing of knowledge is one of the attractions of collaborative learning, but what is also drawing attention is the fact that it enables the participants to have "experiences of constructing interhuman social relationships" such as the joy and difficulty of communicating with others in its process, which would be impossible just by learning alone. Consequently, collaborative learning is expected to bring about better educational results [1][2][3].

2 Conventional Learning

2.1 Conventional Collaborative Learning

In the field of education, from primary to higher levels, various forms of collaborative learning have been attempted thus far. In Japanese primary education, for example, pupils engage in group-based study projects, such as observation of living organisms, cooking practice, exploration of history, and investigation of their cities and towns, while cooperating and sometimes sharing tasks with each other. In higher education, on the other hand, efforts are being made to provide students with diverse collaborative learning experiences including language study, science experimentation, and development of computer application software and robots. In particular, educators involved in collaborative learning in higher education often serve as prompters of students' learning, only making a few comments on their presentations, whereby the students play a main role in classes so that their sense of independence can be nurtured. Although fields of study vary widely as described above, there have been an increasing number of instances of collaborative learning programs implemented under various circumstances as their effects to boost educational efficiency have been substantiated. In addition, starting from 2011, the Japanese Government is scheduled to extend full-fledged financial support to educational institutions that are actively engaged in collaborative learning.

2.2 Issues in Conventional Learning

Regardless of the content, collaborative learning has been practiced under various circumstances. However, it is possible that practicing collaborative learning becomes difficult depending on the content of study In the case of learning programming by using a computer language, for instance, if a group is required to develop one computer application, it is difficult to complete the application while consolidating the ideas of each member in the course of learning since it is the norm in the recent education environment that everyone is provided with his/her own computer. If the application is developed under such an education environment, there will be more occasions on which each member studies on an individual basis, compiling and debugging the program on his/her own computer, than ones where the members cooperate with one another. As a result, the efficacy of collective learning will highly likely be compromised. Suppose that each group is asked to share a single computer in order to avoid such a situation, it may be feasible to work in collaboration if the group consists of only three members who will surround the computer, but if the number of the members exceeds that, chances are high that the development of the application will be impeded. The author understands empirically that even if the task is divided and shared separately by members in charge of programming and those in charge of inputting the program, it happens more often than not that the program conceived by the former is misinterpreted by the later after verbal explanation. Misinterpretations of the information conveyed from member to member necessitate corrections of the program, thereby leading to a waste of time. Moreover, the fact that they are unable to make themselves understood accurately can give rise to frustration among them, thus negatively affecting the progress of learning. Therefore, it is safe to conclude that implementing a study program for computer programming in the form of collective learning is difficult.

Meanwhile, an application development method called pair programming is similar to the aforementioned procedure in that it entails collaborative work in computer programming [4][5]. This method places emphasis on the aspect of complementary work in task sharing. If learners with a certain knowledge of computer programming work together in accordance with this methodology, improvement of programming efficiency can be expected. However, it is extremely difficult for beginners in programming, who lack in specialized knowledge, to carry on with their necessary work based on this technique. In one of the classes of which the author has taken charge, it was necessary to instruct law students with no knowledge of programming. After a series of experiments in instruction methodology, the author concluded that it is imperative to develop a system that supports collaborative learning as described below: a system in which multiple learners share a single desktop screen on a software basis and are permitted to input their ideas from the keyboard whenever they come up with good ones, which are in turn evaluated by other learners in order to advance the process of collaborative work. This system relies on an application of the function of UltraVNC, a program capable of displaying the desktop screen of a remote computer on your own computer screen, modified for the ease of use by students. The details of the system are explained in the following section.

3 Support for Collaborative Learning through Sharing of a Desktop Screen

3.1 Overview

The system which we developed is working as a part of "FUTURE (Fukuoka University Telecommunication Utilities for Research and Education)" built by Fukuoka University. The system developed in this study has a very simple configuration, in which one group is supposed to comprise five students. For each group, five tablet computers are prepared, with one allocated to each of the members. These five tablet computers are interlinked via an IP network for communication. Table 1 shows the hardware and software configuration of the computers.

In addition, the roles of the five tablet computers are divided into two: one functions as a server, and the other four serve as clients. The tablet computer functioning as a server provides its desktop screen for multiple users, whereas the other client computers access the server and display its desktop screen on their screens, whereby one desktop screen is shared by multiple users. Fig.1 shows the schematic image of this process.

Table 1 Hardware and software configuration

Hardware	
Vendor:	Lenovo
Model:	X200 Tablet
CPU:	Intel Core2 Duo 1.86GHz
Memory:	4GByte
HDD:	256GByte
Software	
OS:	Microsoft Windows 7 Enterprise
Other:	UltraVNC Ver. 1.0.9.5
	CWSS (Collaborative Working Support System)

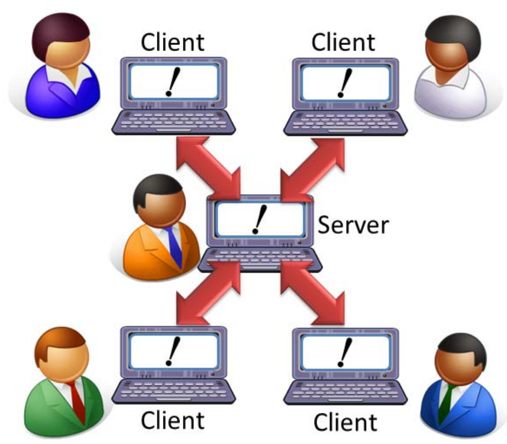


Fig.1 Schematic image

3.2 Sharing Procedure

Fig. 2 shows the procedure for sharing a desktop with multiple users. In principle, server and client programs of UltraVNC are run on each tablet. However, since simply starting such programs is not friendly for students, this system utilizes "CWSS (Collaborative Working Support System)," a supportive application developed by the author that consists of two programs: "CWSSS (Collaborative Working Support System Server)" and "CWSSC (Collaborative Working Support System Client)." First, the CWSSS is executed on the server to activate the UltraVNC server, during which process the CWSSS generates a password for accessing the UltraVNC server and associates it with the UltraVNC. Following the completion of the password setup, the CWSSS displays the IP address and access password of the tablet on which the UltraVNC server is in operation. After the server is ready, the CWSSC is executed on the client computers to launch the UltraVNC clients. At this point, the CWSSC prompts the users to input the IP address and access password of the UltraVNC server. Therefore, the client users need to be apprised of the information necessary for accessing the UltraVNC server by the user of the tablet on which the server is in operation. The input of the necessary details by the users on the CWSSC screens allows the UltraVNC clients to receive information for accessing the UltraVNC server, and the clients then start displaying the desktop screen of the server on their own screens, thereby realizing the sharing of the desktop. Access passwords are generated in order to prevent unauthorized persons from gaining access to the UltraVNC server without permission. Accordingly, the passwords must not be disclosed to any outside party if the confidentiality of collaborative learning needs to be maintained. The passwords set up on the UltraVNC server are changed forcibly every time the CWSSS is launched. This makes it impossible to reuse the same passwords, thus guaranteeing a high level of security.

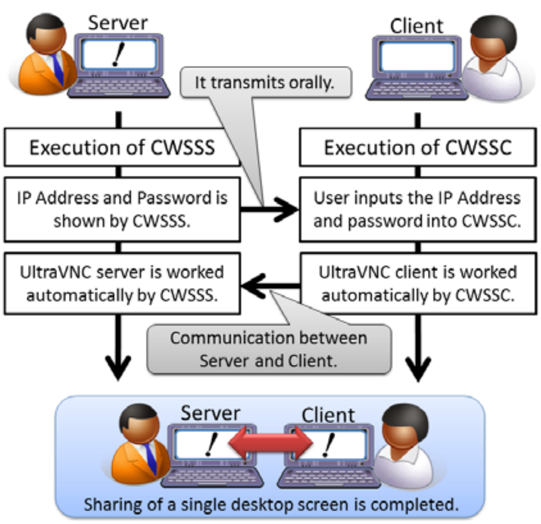


Fig.2 Procedure for sharing a desktop

3.3 Actual Practice of Collaborative Learning through Sharing of a Desktop Screen

We actually implemented a collaborative learning program in which desktop screens were shared. Table 2 shows the details of class.

Table 2 Details of class

Subject:
computer programming
Number of participants:
30 (divided into 6 groups)
Class duration:
90 minutes
Class frequency:
14 times
Implementation period:
October 2010 to January 2011 (4 months)
Participants' capabilities and knowledge:
Internet surfing Document creation on Microsoft Word
Used computer language:
HSP (Hot Soup Processor)
Class participants:
Law students in their sophomore and junior year
Class objective:
Develop a simple program
Example of the program to develop:
Image file viewer program Movie player program Webpage downloader program

The computer language used in the classes is called HSP (Hot Soup Processor), a BASIC-like language developed in Japan. With its simple grammar, the language is less overwhelming to students who study a computer language for the first time and easier for them to learn. In the classes, the students, using HSP, developed programs for displaying still images, playing movies, downloading HTML files by accessing websites, etc. Fig.3 and Fig.4 show how the students were working on their tasks.

Fig.3 The situation of exercise (1/2)

Fig.4 The situation of exercise (2/2)

4 Results Obtained by Using the System

The evaluation of the results of this study was done in terms of the following three aspects: "work efficiency," "communication efficiency of among the students," and "comprehension of the course." The sections below compare the instances where desktop screens were shared and those where they were not in the actual collaborative learning program. The students were presented with a task to develop a program in every class.

4.1 Work Efficiency

Although an accurate comparison is impossible as we did not measure the exact time, more than half the groups whose members did not share a desktop screen could not complete their programming work even past the finish time. By contrast, almost all the groups whose members shared a desktop screen were able to finish their programming by the end of a class. It has therefore been demonstrated that sharing a desktop screen contributes to improvement of work efficiency.

4.2 Communication Efficiency among the Students

When a desktop screen was not shared, the students found it difficult to understand the opinions and ideas of others. This difficulty was attributable to their inability to convey necessary information both thoroughly and accurately, as well as to the tardiness of communication. When a desktop screen was shared, the students realized that it was easier to accurately convey their opinions to others through verbal explanation and at the same time by describing or depicting their conceptions with keyboards and mouse devices in a concrete and precise manner. Furthermore, in the "questionnaire survey on the course" conducted at a later date, more than half the students commented that they "felt the joy and efficacy of collaborative work making use of this system and would like to see this applied to collaborative learning in other subjects as well."

4.3 Comprehension of the Course

When a desktop screen was not shared, the students had to deal with a host of complicated and time-consuming chores in the course of program development, thus resulting in a squandering of time that could have been spent on doing more essential tasks. Consequently, it was pretty hard to heighten the understanding of the classes. When a desktop screen was shared, on the other hand, the burdens of such complicated chores were mitigated, thus enabling the students to expend more of their resources on the study itself. Moreover, with the communication among the students facilitated, they were able to hold in-depth discussions, which improved their comprehension of the course.

5 Conclusions

In this manuscript, the author has discussed the actual practice of collaborative learning in computer programming and its efficacy. An advantage of the environment proposed in this manuscript is the extreme simplicity of its system in which generic personal computers are interlinked with each other via an IP network, which enables implementation of collaborative learning programs not only in a single classroom but also between remote areas. Combining this system with multi-point connection teleconference systems, which have been seeing widespread use recently, is expected to allow the construction of a more powerful environment for collaborative learning; therefore, there is still room for advancement in this research field. Since the system entails multiple users sharing the same desktop screen, they need to be careful not to interfere with the operations by the others so that this does not presents a disadvantage. It may be necessary to take some sort of measures to prevent a simultaneous operation by multiple users, e.g. developing a technology or establishing rules with the intention of suppressing or preventing such interference. Fukuoka University, to which the author belong, has introduced a “Meeting Perfe” [6] paperless meeting system, which enables the sharing of PowerPoint and PDF files by multiple persons through the synchronization and simultaneous display of file pages. We are now cogitating on how our system can be utilized effectively in parallel with this one.

References:

[1] Akiko Inaba, and Toshio Okamoto: A Group Dialogue Model for Distributed Cooperative Learning System, *IEICE technical report. Education technology*, Vol. 94, Issue. 359, 1994, pp. 47-52.

[2] Jeremic, Z., Jovanovic J., and Gasevic D.: Project-based Collaborative Learning Environment with Context-aware Educational Services, *Proceedings of the 4th European Conference on Technology Enhanced Learning (ECTEL09)*, LNCS 5794, Springer-Verlag, Sept/Oct 2009, pp.441-446.

[3] Robert J. Harris: Developing a collaborative learning environment through technology enhanced education (TE3) support, *Education + Training*, Vol. 50 Iss: 8/9, 2008, pp.674 -686

[4] Arisholm, E., Gallis, H., Dybå, T., and Sjøberg, D.I.K. "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise, *IEEE Transactions on Software Engineering*, Vol.33, No. 2, 2007, pp 65-86.

[5] Laurie Williams: Integrating Pair Programming into a Software Development Process, *Proceedings of the 14th Conference on Software Engineering Education and Training*, 2001, p.27.

[6] NTT Advanced Technology Corporation: Meeting Perfe (in Japanese), http://www.ntt-at.co.jp/page.jsp?id=1793&content_id=883

Integrating MyItLab in an Introductory Computer Applications Course

Gina Cooper, PhD
Devry University, Columbus, OH, USA

Abstract - *Introductory computer courses experience large attrition rates, and are faced with many challenges due to the wide range of incoming student abilities and student learning styles. Computer assisted learning has been shown to be an effective tool for increasing student comprehension [1][2]. Computer Basics 101 is an introductory computer applications course at a regionally accredited for-profit university. In Spring 2009, a computer based training program called MyItLab was introduced to improve student comprehension and satisfaction. Three areas were studied to determine the success of the computer assisted learning module: student satisfaction based on end of semester surveys, grades, and student retention. Statistics were generated for one instructor's Computer Basics 101 courses prior to introducing MyItLab and after introducing MyItLab to determine if there was a significant improvement in overall student outcomes. This paper explains the findings of this research as well as identifies advantages and disadvantages to MyItLab.*

Keywords: Assessment, Introductory Course, Online learning, Web-based instruction

1 Introduction

An introductory computer applications course at any University is faced with several challenges. Students enter the course with varying degrees of prior knowledge and comfort with the applications. Computer Basics 101 is an introductory computer applications course at a regionally accredited for-profit university. The course covers computer hardware basics, Internet concepts, Microsoft Word, Microsoft PowerPoint, and Microsoft Excel. This course is a prerequisite for subsequent courses in almost every major, thus it is generally taken within the first three sessions by new students. MyItLab, a computer assisted learning tool, was introduced to improve student satisfaction and reduce the course's high attrition rate.

Research has shown that students can learn effectively using computer based instruction and retain information well [1][2]. MyItLab is a computer based training and assessment solution produced by Pearson Education [3]. Both the training and assessment in MyItLab are conducted online using simulations of Microsoft Office products. The training is self-paced and provides feedback to students as they are completing the training to prepare them for the exams.

This online methodology was applied to Computer Basics 101 at the university's Columbus, OH Campus in Spring 2009. Since the university has a geographically diverse population, MyItLab training and support for faculty was centralized by an offsite administrator. A course code is given to faculty members each session. Students use the course code when logging into the website so that their assessment training can be tracked by the faculty member teaching the course. The course codes and the course content are set up prior to the term by a coordinator for all faculty across the university's campuses.

To determine the effectiveness of the computer based training in Computer Basics 101, three areas were examined. End of semester student surveys, grades, and retention were the three factors considered by this study. Data was gathered for five sessions prior to using MyItLab and four sessions after introducing MyItLab. To determine if there existed a significant difference in student satisfaction before and after MyItLab was introduced, a t-test was performed. The statistical analysis shows that a significant improvement was found in student satisfaction with the introduction of MyItLab. MyItLab also improved student grades and retention in Computer Basics 101. This paper describes the findings and extrapolates on methods to improve the course.

2 Initial Course Design

The Computer Basics 101 course is one of the first courses that new freshmen take at the university in Columbus. This course is taught in a lab/lecture classroom in which each student has a computer at his or her desk. This format is successful for both standard lecture classes and computer based training. Computer Basics 101 is a 2 credit hour class and the course description is below:

This course introduces basic concepts and principles underlying personal productivity tools widely used in business such as word processors, spreadsheets, email, and web browsers. Students also learn basic computer terminology and concepts. Hands-on exercises provide students with experience in use of PC's and current personal productivity tools.

Prior to Spring 2009, the course was taught using standard lectures, in-class exercises, lab assignments, exams, and lab exams. The book for the course has been the Exploring Microsoft Office series published by Prentice Hall for all versions of Microsoft Office. Class met once a week for a two hour lecture in which students learned the course concepts and completed several in-class exercises on their own computers to reinforce their learning. These in-class exercises were graded which encouraged attendance.

Following the lecture was a designated lab time in which the students could complete the lab assignments. However, many students did not complete the assignments during lab time and instead completed them as homework. These assignments generally involved modifying partially completed documents, spreadsheets, or presentations or creating new files. The lab assignments challenged the students to apply the skills they learned in the lecture. Some assignments were from the book while others were generated to supplement the book. Once completed the students would submit the assignments electronically to a dropbox on the online learning platform.

A midterm and final exam were given to the students to assess their knowledge. These exams consisted of part multiple choice and part lab exams. This educational technique was a more traditional method of teaching students the concepts of the course.

3 Introduction of MyItLab

The introduction of MyItLab changed the course assignments significantly. The two hour lecture with in-class exercise points remained however the lab time was spent using MyItLab rather than previous lab assignments.

3.1 MyItLab Overview

MyItLab was created by Pearson Education to provide assessment and training for students that is integrated with the textbook. The product is a realistic simulation of Microsoft products providing hands-on training to students. Figure 1 shows an example of a MyItLab screenshot.

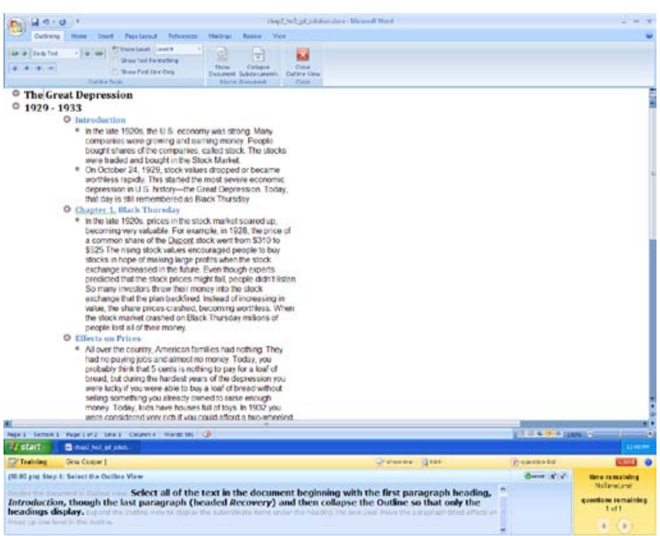


FIGURE 1: MYITLAB TRAINING SCREENSHOT

Each training module has a set of directions listed below the simulation screen. Students have several attempts to complete the training and can ask for a hint or even have the system show them the correct answer before they complete it themselves.

After the student completes a number of training modules, he or she can take the exam in MyItLab. The exams are in the same format as the training modules, however there are a limited number of attempts and the hints are not available. The student immediately receives the result of the exam based on criteria set up by the instructor. For example, all MyItLab exams are set up such that if the student receives 80% or higher a "passed" message is displayed for the student.

The instructor can modify the questions, reset the exam, change the number of attempts and view the solutions to the exam questions. A grade book is also available in MyItLab for the instructor to manage and export grades to a variety of formats.

3.2 MyItLab Implementation

MyItLab training modules are connected to the textbook used for Computer Basics 101. Therefore, chapter training and exams were set up to match the chapter coverage in lectures. The modules set up in MyItLab were Office Fundamentals, Word Chapters 1-3, PowerPoint Chapter 1, Excel Chapters 1-3, and the Final Exam. Students were permitted to take the chapter exams 3 times and the best score would be recorded, whereas the comprehensive final exam could only be taken once.

In addition to the MyItLab exams, three labs were given to the students covering the topics of email, the Internet, and Windows Fundamentals. These labs, along with in-class exercises were the only points in the course outside of MyItLab and only accounted for 13% of the total grade.

MyItLab has several advantages for Computer Basics 101 students. The main benefit is the self-paced nature of the training and assessment. Students who are familiar with Microsoft products can complete the entire course in less than a semester. Those who struggle can spend as much time as they need with immediate feedback from the program in the absence of the instructor. MyItLab allows students to work through the course at their own convenience and at their own level.

Another advantage of MyItLab is that it can be used with only a web browser and internet connection. Students are not required to purchase Microsoft Office in order to use MyItLab. The program requires Windows OS (although some students were able to run MyItLab using a Mac with BootCamp). It also requires Internet Explorer 7.0 or 8.0 with a broadband connection recommended [3]. Students were also required to purchase MyItLab in addition to the book. The extra cost associated with MyItLab was a drawback for some students.

Moving from a traditional lab based model to online assessment was reasonably seamless. Few challenges were encountered with MyItLab. The program is very accurate. It accounts for multiple ways of completing tasks in Microsoft products and is virtually error-free. One challenge faced by several international students was the wording of the exam questions. This issue only affected a small percentage of students and the instructor was easily able to clarify the meaning of the questions for those students. Students who were frustrated with the results of MyItLab often realized it was their own error rather than an error with the program.

4 Results and Discussion

Three areas of student achievement were studied to determine MyItLab's impact on Computer Basics 101. These areas were end of semester student surveys, grades, and retention. For the purpose of this study, retention is defined as those students who completed the course with a passing grade (A, B, C, or D).

Each session, online surveys are administered to students. These surveys are optional, yet students are encouraged to complete them to give the instructor feedback on the course. The survey consists of 26 questions and a comment section. Eight out of 26 questions were singled out as providing the most information for student success. Analysis was performed to determine if there was a statistically significant difference in student evaluation responses prior to using MyItLab and after using MyItLab. Students rated their degree of knowledge from 1 to 4 with the following labels 1= poor, 2 = fair, 3 = good, 4 = excellent. The fifth question is rated 1-12. These eight questions were paraphrased below:

1. Course assignments promoted learning.
2. Course was interesting and challenging.
3. Textbook and lab materials were used effectively.
4. Supplemental material such as audio-visual aids and lab activities were valuable.
5. Instructor's teaching effectiveness.
6. New knowledge and skills acquired in the class.
7. Overall course rating.
8. Instructor's ability to assist students with achieving the objectives of the course.

Averaged data is illustrated in Table 1 for the eight questions administered in the survey. 35 data points were used from surveys for those students prior to using MyItLab and 28 data points were collected from those students after MyItLab was introduced. A t-test was performed on the data with a null hypothesis of "the mean student evaluation data for students not using MyItLab is equal to the mean for students using MyItLab." As question 5 is variable, it was not included in the t-test. The t-test results yielded a p-value of 0.0002. This indicates that the null hypothesis can be rejected with an alpha of 0.05, showing that there is a statistically significant difference in the means of the student evaluations between the students using MyItLab and those not using MyItLab in Computer Basics 101. The mean student satisfaction value from students prior to using MyItLab was 3.632 and after using MyItLab the mean value was 3.846. These results show a 6% increase in student satisfaction when using MyItLab.

These results are noteworthy since student satisfaction can lead to more motivated learners, higher grades, and better persistence. From the raw data in Table 1, several scores of 4 were given by students indicating a high degree of satisfaction with the course.

TABLE 1
STUDENT EVALUATION RESULTS BEFORE AND AFTER MYITLAB INTRODUCTION

	Before MyItLab					After MyItLab			
Question	SU 07	SU 07	SU 08	SU 08	SU 08	SP 09	SP 09	SU 09	FA 09
1	3.83	3.75	3.87	3.75	3.4	3.9	3.8	3.8	3.8
2	3.75	3.75	3.73	3.42	3.2	4	4	3.6	4
3	3.92	3.75	3.67	3.75	3	3.9	4	3.8	3.9
4	3.75	3.75	3.87	3.83	3.2	4	4	3.8	3.9
5	11.6	12	11.7	11.7	11.8	12	12	11.8	12
6	3.67	3.75	3.47	3.67	3	3.7	3.8	3.8	3.5
7	3.83	3.75	3.53	3.5	3.2	3.8	4	3.8	3.5
8	3.67	3.75	3.87	3.92	3.6	4	4	4	3.6

In addition to student satisfaction surveys, student retention and grades were also reviewed. Grades as well as the number of students who dropped the course were compiled for students taking Computer Basics 101 before and after MyItLab was introduced. Table 2 illustrates these results.

In the table, "percent completed" refers to the students who did not drop the course. These students may or may not have passed Computer Basics 101, but they completed the course. These values were similar regardless of whether or not MyItLab was part of the course. This indicates that approximately 7-8% of students drop Computer Basics 101 regardless of the course content, most likely due to outside circumstances. Percent passed includes all students receiving a grade of A, B, C, or D divided by the total number of students completing the course (excluding students who dropped the course). Courses using MyItLab had a six percent higher pass rate than those who do not use MyItLab. Likewise, 5.6% more students failed Computer Basics 101 when MyItLab was not used as part of the course.

TABLE 2
STUDENT PASS RATE AND RETENTION BEFORE AND AFTER MYITLAB

	Percent Completed (including F's)	Percent Passed (grades A-D)	Percent Failed
Before MyItLab	92.28%	87.04%	11.77%
After MyItLab	92.81%	93.20%	6.17%

Based on the three factors of student success, MyItLab has had a positive impact on Computer Basics 101. Students were more satisfied with the course; more students passed the course, therefore leading to higher retention. The same number of students dropped the course whether or not MyItLab was part of the course but these students most likely dropped due to outside factors.

5 Course Improvement

The results from this study indicate that the inclusion of MyItLab in Computer Basics 101 led to increased student satisfaction, retention, and higher grades. While these results are significant, this course can still be improved to increase student comprehension.

While the simulator is a great training tool, some faculty expressed concerns that the skills learned using the simulator would not carry over to use of the actual Microsoft Office applications. An area of further study would be determining the extent to which skills learned in the simulator correspond to student skills in the actual applications. Students would need proficiency with the actual applications for real-world problems rather than responding to training and test questions. Developing their own documents, spreadsheets, and presentations without the assistance of tools such as the simulator are a necessary skill for the workforce.

The inclusion of an end of term course project that merges Word, Excel, and PowerPoint would challenge the students and provide an opportunity for them to use the knowledge they gained through MyItLab in the actual applications. This project could require the students to research values (prices of computers for example) online and input the data into Excel. They could create formulas and charts to indicate the best option then copy their results to Word for a professional memo. Finally they could create a PowerPoint presentation to display their findings. Such a project would reinforce the student's learning in the course.

Another improvement to the course could be additional labs to be completed in the applications. Students could complete the training and exams in MyItLab and complete extra lab assignments to reinforce their knowledge with Microsoft Office products.

While the course already has in-class exercises, an additional exercise such as an in-class lab or group project can provide teamwork skills as well as require the students to solve a business problem in a specific time frame.

These options for course improvement provide opportunities for students to test the knowledge they gained from lecture, in-class exercises, and the MyItLab training and assessment. A combination of MyItLab and additional assignments within the Microsoft Applications will be implemented and their impact on student satisfaction, retention, and grades analyzed.

6 Conclusions

Introductory Computer Applications courses suffer from high attrition rates, low grades and student dissatisfaction. To improve student success, MyItLab was introduced in Computer Basics 101 at a regionally accredited for-profit university. Data was analyzed for the Computer Basics 101 before and after MyItLab was introduced to determine the success of its online training and assessment. Three areas were reviewed for student success. These were student satisfaction from end of semester surveys, student retention, and grades. A t-test was performed to determine whether there existed a statistically significant difference between the responses of students before and after MyItLab was introduced. Based on the data, the t-test showed that students were more pleased with Computer Basics 101 when MyItLab was part of the course. In addition, 93% of students passed Computer Basics 101 when MyItLab was used compared to only 87% before MyItLab. Failure rate was virtually halved when MyItLab was introduced. These results indicate that students are happier and more successful when using MyItLab. To challenge students and improve their comprehension of the material, additional labs and projects were explored and explained in the Course Improvement section. Based on the findings of this analysis, using MyItLab for training and assessment was successful for Computer Basics 101 at the university.

7 References

[1] Kulik, C. and Kulik, J.. "Effectiveness of Computer Based Instruction: An Updated Analysis". Computers in Human Behavior, v. 7 issue 1-2, 1991, pp. 75-94.

[2] Wenglinsky, H. "Does it compute? The relationship between educational technology and student achievement in mathematics." Educational Testing Service Policy Information Center. 1998.

[3] Pearson Education. http://www.myitlab.com. 2011.

National Instruments® LabVIEW™: Ultimate Software for Engineering Education

Olfa Boubaker
National Institute of Applied Sciences and Technology, Tunis, Tunisia

Abstract - *National Instruments® LabVIEW™ is considered, nowadays, as the most professional software, among others, for data acquisition, data analysis, real time and remote control. This paper presents, in details, the key motivations for the use of such virtual reality programming environment by many prestigious universities and explains the main reflections on how this software gives a valuable application for engineering training. A bibliographical review of numerous applications in a general framework of engineering education will be presented and classified according to different engineering disciplines. Several experiences dealing with remote control laboratories are also summarized with emphasis on the practical design implementation. The trends of introducing a course within the new curriculum of the National Institute of Applied Sciences and Technology at Tunisia, aiming to improve the technical skills of engineering students and enhance their recruitment criteria, will be finally addressed.*

Keywords: Engineering education, LabVIEW™ software, LabVIEW™ hardware, virtual reality, web-based remote control.

1 Introduction

The evolution of engineering education is related to advances on technology, theoretical design methods and their real-time implementation [1]. It is also important to note that the evolution of the engineering theory is closely related to the engineering education [2]. The next generations of students must then receive the scientific and pedagogical supports required to verify conventional techniques, develop new tools and verify their real implementation. Computing and communication technologies are useful tools to improve such educational strategies. They can provide to the engineering education system a significant impact. However, the quick rate of change in the fields of technology poses particular problems for engineering institutions. Customized engineering test systems have in many times a number of limitations such as inflexible and unfriendly programming structure, excessive cost, hardware limitations, and incompatibility issues [3]. To realize more efficient and more user-friendly systems without using very costly custom-written software and tools, the key solution is to integrate in engineering institutions virtual and remote laboratories [4].

National Instruments® LabVIEW™ software/ hardware is considered nowadays as the most efficient programming environment, among others, for virtual reality model design, data acquisition, data analysis, instrument control and also web-based remote control. Assuming that the most of the engineering problems deal with physical variables such as temperature, speed, position, current, voltage, pressure, force, torque... LabVIEW software with its suitable of data acquisition system gives us a visual look to all these quantities [5]-[6]. Furthermore, most of experiences can be made available to the remote area user via Internet link or satellite link [7].

This paper presents the key motivations for the use of that system and explains, in details, the main reflections on how such software/hardware environment gives an effective and efficient application for engineering training.

The paper will be organized as follows: In the second section LabVIEW software/Hardware environment will be presented. A bibliographical review of different applications using LabVIEW programming environment in a general framework of engineering education will be presented and classified according to different disciplines in the third section. The trends and the perspectives of introducing a course within the new curriculum of the National Institute of Applied Sciences and Technology (Tunisia) aiming to prepare students to be National Instruments certified will be finally addressed in the section IV.

2 Software/Hardware Environment

LabVIEW™ system is a data acquisition and programming environment that allows flexible acquisition and processing of analog and digital data [8]. The LabVIEW environment [9] contains a wide-ranging set of tools for acquiring, analyzing, displaying, and storing data. LabVIEW software can be used to communicate with hardware devices such as data acquisition, vision, motion control devices, GPIB, PXI, VXI, RS-232, and RS-484 devices. LabVIEW has also built-in features for connecting applications to the Web using the LabVIEW Web Server and software standards such as TCP/IP networking and ActiveX.

The main characteristic that distinguishes LabVIEW from other data acquisition programs is its highly modular graphical programming language, "G," and a great library of functions [8].

LabVIEW environment is famous for its flexibility, reusability, self-documentation modularity, multi-platform portability, and compatibility with existing code and with different interface hardware [3]. LabVIEW software is distinguished by its extendible libraries and its multi-advanced debugging features. It is also characterized by an intuitive Graphical User Interface (GUI) and Multimedia capabilities for future developments. Furthermore, it is possible to create stand-alone executables and share libraries.

2.1 LabVIEW Software

LabVIEW programs [9]-[13] are called virtual instruments (VIs) because their appearance and operation imitate physical instruments. A LabVIEW virtual instrument consists of a front panel which is considered as a user interface, a block diagram which is considered as a LabVIEW program and a connector pane which is a set of terminals on the icon that corresponds to the controls and indicators of a VI, similar to the parameter list of a function call in text-based programming languages (see Fig.1).

In the front panel, the user can insert the inputs (sliders, knobs, values entered from the keyboard, push buttons, selector switches) and the outputs (indicators, LEDs, graphs, strip charts, sounds). The Controls palette contains the controls and indicators used to create the front panel. Block diagrams integrate terminals, subVIs, functions, constants, structures, and wires, which transfer data among other block diagram objects. Nodes are objects on the block diagram that have inputs and/or outputs and perform operations when a VI runs. Nodes can be functions, subVIs, or structures. A subVI is similar to a function in a text-based programming language. Every VI displays an icon in the upper right corner of the front panel window and block diagram window [9]. The Functions palette contains the VIs, functions and constants used to create the block diagram.

The most common execution structures are While Loops, For Loops and Case Structures which can be used to run the same section of code multiple times or to execute a different section of code based on some condition. LabVIEW has other, more advanced types of execution structures such as Event Structures used to handle interrupt driven tasks and sequence structures used to force execution order. Shift registers are used in loops to pass values from previous iterations through the loop to the next iteration. Tunnels are used to feed data into and out of structures. In LabVIEW software, it is helpful to group data in arrays and clusters. Arrays mix data of the same data type into one data structure, and clusters join data of multiple data types into one data structure. The progress of data from side to side the nodes determines the execution order of the VIs and functions on the block diagram. LabVIEW programs follow a dataflow model for running VIs. A block diagram node executes when it receives all required inputs. LabVIEW contains some powerful debugging tools.

There are two general types of software bugs that may be used, those that prevent the program from running and those that generate bad results or incorrect behavior [9]-[13].

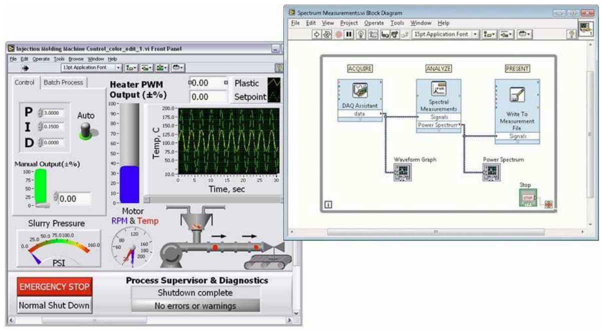


Fig.1. LabVIEW virtual graphical programming environment

2.2 LabVIEW modules and toolkits

The NI LabVIEW Modules and Toolkits are the add-on software tools that extend LabVIEW graphical programming for specific applications. More than 25 LabVIEW modules and toolkits exist [14]. For example, NI Modules and Toolkits created for Embedded Design integrate:

– *Real-Time Module*
– *NI Real-Time Execution TraceToolkit*
– *FPGA Module*
– *Microprocessor SDK*
– *Statechart Module*
– *Mobile Module*
– *DSP Module*
– *Embedded Module for ARM Microcontrollers*

Whereas NI Modules and Toolkits for Control design and Simulation integrate:

– Control Design and Simulation Module
– PID and Fuzzy Logic Toolkit
– Real-Time Module
– Execution Trace Toolkit
– FPGA Module
– Statechart Module
– Simulation Interface Toolkit

2.3 LabVIEW Hardware and Data Acquisition System

LabVIEW integrates many thousands of hardware devices which can be used with a single development environment. In addition to data acquisition hardware [15], further test, measurement, and control hardware are offered [9]. Examples include digital Multimeters, High-Speed Digitizers, RF Signal Analyzers, RF Signal Generators , signal Generators, High-Speed Digital I/O, Switches, Programmable Power Supplies, Reconfigurable FPGA I/O, Motion Controllers and vision Systems…

The components of a Data Acquisition System (see Fig.2) are:

– Transducers/Sensors:

A transducer or sensor converts a physical phenomenon into a measurable electrical signal. Examples of transducers are thermocouples for temperature, strain gages for pressure…

– Signal Conditioning

The signal conditioning device maximizes the precision of a system, gives sensors the ability to work properly, and guarantees safety.

– Data Acquisition Hardware

The data acquisition hardware is an interface device between the computer and the outside world. It digitizes incoming analog signals. It includes analog output, digital I/O, counter/timers, and triggering and synchronization circuitry.

– Driver and Application Software

There are two layers of software in a data acquisition system: driver software and application software. Driver software is the communication layer between the application software and the hardware. The application layer is a development environment in which a custom application is built.

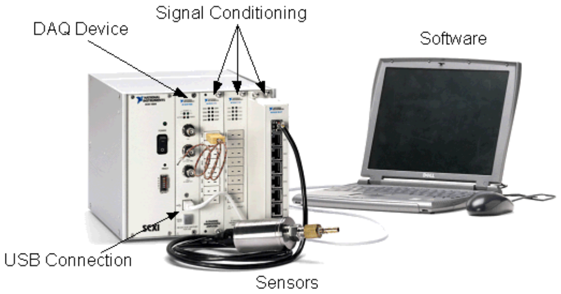


Fig.2. Data Acquisition System

LabVIEW driver software supplies seamless integration across multiple types of instruments, buses, and sensors, including data acquisition devices, boxed instruments, modular instruments, motion controllers, motor drives, machine vision, image processing hardware, wireless sensors, and field-programmable gate arrays (FPGAs) [9].

For example, the NI-DAQmx driver software can join together more than 200 data acquisition devices in LabVIEW on major buses and factors, including USB, PCI, PCI Express, PXI, PXI Express, wireless, and Ethernet.

NI-DAQmx devices include MAX, a configuration and test utility. MAX is used to Configure and test NI-DAQmx hardware and Create NI-DAQmx virtual channels that can be referenced in any programming language. NI-DAQmx virtual channels are used to map configuration information to specified physical channels on NI-DAQmx devices. Virtual channels are used to configure input ranges and signal conditioning parameters, convert sensor voltage measurements into real-world engineering units, add signal conditioning configuration apply user-defined scaling operations and creating NI-DAQmx tasks by adding timing, triggering, and synchronization information [9].

2.4 NI Educational Laboratory Virtual Instrumentation Suite

National Instruments was developed an efficient platform for engineering education used by the leading international companies and universities: the National Instruments Educational Laboratory Virtual Instrumentation Suite (NI ELVIS). The NI ELVIS (see Fig.3) integrate a suite of 12 instruments in one compact form factor [16, 17, 18].

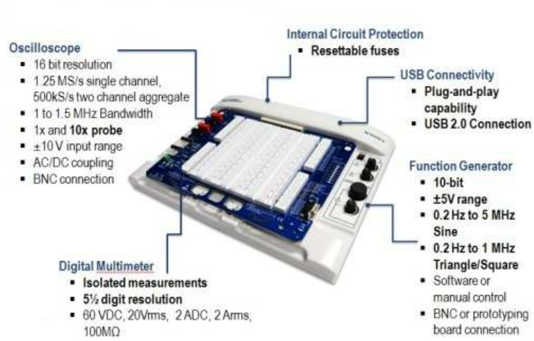


Fig.3. Educational Laboratory Virtual Instrumentation Suite

Many modules are built on the ELVIS platform to teach more specific disciplines, including circuit design, instrumentation, control, microcontroller/embedded systems, mechatronic, telecommunications…

2.5 LabVIEW embedded Platforms

To reduce the cost and shortens time to design, prototype, and deploy embedded systems, it is possible to take the advantage of the NI embedded platforms [19]. There are three deployment technologies:

– Unpackaged Embedded Systems

These systems have a variety of processor architectures to choose from and a small set of operating system and I/O support. The software development tools for such systems are not integrated. The National Instruments NI Single-Board RIO represents an example of an unpackaged embedded system (see Figure 4).

– Packaged embedded systems

These systems are generally more expensive but they often come with integrated software development tools and a more extensive set of integrated I/O options. The National Instruments NI CompactRIO represents an example of an unpackaged embedded system (see Fig. 5).

– Industrial PCs

The PXI platform is an example of an industrial PC architecture. PXI is the open, PC-based platform for test, measurement, and control. PXI provides a variety of processing and I/O options for embedded applications such as multi-core processors, high-resolution DC measurements, digitizers, motion control...

Fig.4. NI Single-Board RIO: an unpackaged embedded system

Fig.5. CompactRIO packaged embedded System

FPGA technology can be used within industrial PCs. All FPGAs within National Instruments products can be programmed with the LabVIEW FPGA Module.

An example of a cost-effective fire suppression system for the main deck of FedEx Express aircraft using LabVIEW and CompactRIO is found in [20].

The LabVIEW Robotics Starter Kit based on the sbRIO 9631 platform, which includes a Freescale MPC5200 processor, 110 digital lines, up to 32 analog outputs, four analog inputs, and 32 industrial 24-V digital I/O is given in [21].

3 LabVIEW: A Multidisciplinary Software

3.1 Multidisciplinary Laboratory Projects

Many prestigious universities and colleges, throughout the world, that deliver undergraduate programmes in science and engineering are currently incorporating virtual instruments as teaching, measurement and analysis tools for student learning. Table 1 shows a bibliographical review of the main applications using LabVIEW programming environment in a general framework of engineering education. The references are classified according to different disciplines.

Table 1. Multi-disciplinary applications of LabVIEW

disciplines	references
Biomedical engineering	[22]
Chemical and biotechnological engineering	[23]
Environmental and ecological engineering.	[24]
Aerospace and Mechanical engineering	[25], [26]
Electro-Mechanical engineering	[27]
Mechatronic engineering	[28]
Robotics	[29], [30], [31]
Control engineering	[32], [33], [34]
Electronic engineering	[35, [36]
Power electronics	[37] [38]
Electrical machines	[39] [40] [41]
Instrumentation	[42]

3.2 Remote Control Projects

Distance education as well as remote experimentation, is a modern field that aims to deliver education to students who are not physically present on their class or their tested instrumentation [43]. Nowadays, many students and educators communicate with the educational material through Internet-based technologies. Remote access is increasingly simplified and information is stored easily. Time savings, sharing resources of expensive equipment and individual access to experimentation are the main factors that motivate remote laboratories over the world [44]-[47]. Some remote pioneered projects using LabVIEW environment based on this idea are described in [48]-[55].

In such LabVIEW environment, the remote users can interact with experiments through the Internet without problems. In the friendly user interface, the distant user can change system parameters and observe system response in textual, graphical, or video format. In addition, some remote laboratories incorporate a booking system, which enables remote users to book experiments in advance [48].

3.3 LabVIEW within the New Curriculum of INSAT

This section discusses the introduction of the LabVIEW-based teaching/ learning tools within the new curriculum [56] of INSAT launched in 2010 year. These activities will be introduced in the second semester of the second year to the three specializations (*Mechatronic, Industrial and Systems Engineering,* Embedded system *Engineering*) offered to the *control and computer science* engineering students. Since LabVIEW™ is an interactive, graphical programming language that can be used to build sophisticated applications in a shorter time, the first objective of this course will be to improve the learning experience of undergraduate engineering students and enhance their interest in a graphical-based computer language for data acquisition, data analysis, and real time control.

Fig.6. Industrial Systems laboratory (INSAT, Tunisia)

The learning methodology consisted of four stages:

– An introductory learning period for the LabVIEW™ programming language and Hardware environment.
– An Application learning period aiming at solving a set of specific software exercises using LabVIEW environment.
– A designing and developing LabVIEW project period aiming to acquire knowledge by direct experiences on Industrial Systems Laboratory (see fig.6).
– One-hour multiple-choice test: the Certified LabVIEW Associate Developer (CLAD) exam to prove the student technical competency.

NI CLAD Certification will demonstrate first that students have the skills to create high-quality applications with NI software platforms. It gives employers confidence in the technical students' abilities to create new business opportunities.

4 Conclusion

LabVIEW is an exclusive programming environment for data acquisition, data analysis, real time and remote control. It gives engineering the ability to program their complex monitoring applications with reducing time for development. Teaching engineering using LabVIEW environment not only improve their learning abilities in control monitoring but also enhance their skills to create high-quality applications. Using such common tools and platforms in engineering education undoubtedly help educators to produce qualified engineers who are ready for innovation in research and industry careers, able to share results and collaborate with international projects. Such an approach can certainly enhance recruitment criteria for engineering job functions and improve the country's technical advantage.

5 References

[1] R. Bars, P. Colaneri, C. E. de Souza, L. Dugard, F. Allgöwer, A. Kleimenov, A. Kleimenov, C. Scherer. "Theory, algorithms and technology in the design of control systems"; Annual Reviews in Control, vol. 30, n°1, 19-30, 2006.

[2] P. Horáček. "Laboratory experiments for control theory courses: A survey"; Annual Reviews in Control, vol. 24, 151-162, 2000.

[3] N. Ertugrul. Towards virtual laboratories: "A survey of LabVIEW-based teaching/ learning tools and future trends"; International Journal of Engineering Education, vol. 16, n°3, 171-180, 2000.

[4] B. Balamuralithara, P. C. Woods. "Virtual laboratories in engineering education: The simulation lab and remote lab"; Computer Applications in Engineering Education, vol. 17, n°1, 108-118, 2009.

[5] T.L. Schwartz, B.M. Dunkin. "Facilitating interdisciplinary hands-on learning using LabVIEW"; International Journal of Engineering Education, vol. 16, n°3, 218-227, 2000.

[6] P. Tiernan. "Enhancing the learning experience of undergraduate technology students with LabVIEW™ software"; Computers and Education, vol. 55, n°4, 1579-1588, 2010.

[7] C. Salzmann, D. Gillet, P. Hugeuenin. "Introduction to real-time control using LabVIEW™ with an application to distance Learning"; International Journal of Engineering Education, vol. 16, n°3, 218-227, 2000.

[8] C. J. Kalkman. "LabVIEW: A software system for data acquisition, data analysis, and instrument control"; Journal of Clinical Monitoring, vol. 11, n°1, 51-58, 1995.

[9] (2010) The National Instruments® LabVIEW™ Corporation website. [Online]. http://www.ni.com/labview/

[10] "LabVIEW™ Core 1 Course Manual". National Instruments Corporation, 2009.

[11] "LabVIEW™ Core 2 Course Manual". National Instruments Corporation, 2009.

[12] "LabVIEW™ Core 1 Exercices". National Instruments Corporation, 2009.

[13] "LabVIEW™ Core 2 Exercices". National Instruments Corporation, 2009.

[14] The National Instruments® LabVIEW™ Toolkit website. [Online]. Available: http://www.ni.com/labview/family/

[15] "Labview : Data Acquisition Basics Manual". National Instruments Corporation, 1998.

[16] J. Del Rio, E. Trullols, A. Manuel, J. Mendez, G. Prados, R. Palomera. "ELVIS. A new tool for teaching and training"; Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference, vol. 2, 1282-1285, 2004.

[17] B. Paton. "Introduction to NI ELVIS, NI ELVIS II, Multisim, and LabVIEWTM Course". National Instruments Corporation, 2009.

[18] "Integrated Suite of 12 Instruments for Interactive, Cost-Effective, Multidisciplinary, Hands-On Learning". National Instruments Corporation, 2008. www.ni.com/pdf/products/us/cat_nielvisii.pdf

[19] (2010) National Instruments LabVIEW Embedded System Design website. [Online]. Available: http://www.ni.com/embedded/

[20] J. Snow, T. Ingram. "Expres delivery, New Electronics"; vol. 42, n°7, 17-18, 2009.

[21] B. Wong. "Tools turn robot projects into child's play". Electronic Design, vol. 58, n°4, 14-15, 2010.

[22] W. D. Smith, G. B. Williams, R. Berguer and J. T. Anderson. "LabVIEW™ facilitates interdisciplinary team projects in graduate biomedical engineering courses"; International Journal of Engineering Education, vol. 16, n°3, 234-243, 2000.

[23] N. Mkondweni, R. Tzoneva and S. Harisson. "Integrating LabVIEW capabilities for monitoring and supervisory control with a B-braun biotech gmbh unit for fermentation processing and direct control"; International Journal of Engineering Education, vol. 21, n°1, 63-74, 2005.

[24] D. Krehbiel, R. Zerger, J.K. Piper. "A remote-access LabVIEW-based laboratory for environmental and ecological science"; International Journal of Engineering Education, vol. 19, n°3, 495-502, 2003.

[25] P. Strachan, A. Oldroyd, M. Stickland. "Introducing instrumentation and data acquisition to mechanical engineers using LabVIEW™"; International Journal of Engineering Education, vol. 16, n°4, 315-326, 2000.

[26] F.K. Lu, P.K. Panicker, M.B. Webb. "Introducing modern laboratory experiences to mechanical and aerospace engineering students"; Proceedings of the ASME International Mechanical Engineering Congress and Exposition, vol. 7, 443-452, 2008.

[27] R. Alba-Flores, D. Hunt. "Incorporating LabVIEW to enhance the learning experience in the electromechanical analysis laboratory"; Proceedings of the ASEE Annual Conference and Exposition, 2008.

[28] M. Ghone, J. Wagner. A multi-disciplinary mechatronics laboratory"; Proceedings of the of American Society for Engineering Education Conference, 1607-1614, 2003.

[29] B. Erwin, M. Cyr, C. Rogers. "LEGO engineer and RoboLab: Teaching engineering with LabVIEW from kindergarten to graduate school"; International Journal of Engineering Education, vol. 16, n°3, 181-192, 2000.

[30] J.M., Gomez-de-Gabriel, A. Mandow, J. Fernandez-Lozano, A. Garcia-Cerezo. "Using LEGO NXT mobile robots with LabVIEW for undergraduate courses on mechatronics"; IEEE Transactions on Education, vol. 54, n°1, 41 - 47, 2011.

[31] V. Wilczynski, J. S. Mittelman, N. Lim. "2,000 robotic applications using the national instruments CompactRIO embedded control system"; Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications, 186-194, 2009.

[32] J.P. Keller. "Teaching PID and fuzzy controllers with LabVIEW". International Journal of Engineering Education, vol. 16, n°3, 202-211, 2000.

[33] S. Daniels, D. Harding, M. Collura. "Introducing feedback control to first year engineering students using LabVIEW"; Annual Conference and Exposition: The Changing Landscape of Engineering and Technology Education in a Global World, 8869-8880, 2005.

[34] P. Pratumsuwan, S. Thongchai, S. Tansriwong. "A fuzzy logic control of electro-hydraulic systems laboratory experiment in mechatronics education"; Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, 149-154, 2010.

[35] Y. Zhou, J. J. Jiang, S. C. Fan. "A LabVIEW-based, interactive virtual laboratory for electronic engineering education"; International Journal of Engineering Education, vol. 21, n°1, 94-102, 2005.

[36] R. Kiliç, B. Karauz. "Chaos training boards: Versatile pedagogical tools for teaching chaotic circuits and systems"; International Journal of Engineering Education, vol. 24, n° 3, pp. 634-644, 2008.

[37] J. M. Jiménez-Martínez, F. Soto, E. de Jódar, J. A. Villarejo, J. Roca-Dorda. "A new approach for teaching power electronics converter experiments"; IEEE Transactions on Education, vol. 48, n°3, 513-519, 2005.

[38] P. Spanik, L. Hargas, M. Hrianka, I. Kozehuba. "Application of virtual instrumentation labVIEW for power electronic system analysis"; Proceedings of the 12th International Power Electronics and Motion Control Conference, 1699-1702, 2007.

[39] N. Patrascoiu. "Modeling and simulation of the DC motor using MatLAB and LabVIEW"; International Journal of Engineering Education, vol. 21, n°1, 49-54, 2005.

[40] D. G. Kasten. "Integrating computerized data acquisition and analysis into an undergraduate electric machines laboratory"; Proceedings of the of Frontiers in Education Conference, vol. 1, T1D 13-18, 2000,

[41] F. S. Sellschopp, M. A. Arjona. "An automated system for frequency response analysis with application to an undergraduate laboratory of electrical machines"; IEEE Transactions on Education, vol. 47, n°1, 57-64, 2004.

[42] A.B. Buckman. "A course in computer-based instrumentation: Learning LabVIEW™ with case studies"; International Journal of Engineering Education, vol. 16, n°3, 228, 2000.

[43] V. Kolias, I. Anagnostopoulos, E. Kayafas. "Remote experiments in education: A survey over different platforms and application fields"; Proceedings of the 11th International Conference on Optimization of Electrical and Electronic Equipment, 181-188, 2008.

[44] P. Bauer, V. Fedák, O. Rompelman. "PEMCWebLab - distance and virtual laboratories in electrical engineering: Development and trends"; Proceedings of the 13th International Power Electronics and Motion Control Conference, 2354-2359, 2008.

[45] P. Bauer, V. Fedak, V. Hajek, I. Lampropoulos. "Survey of distance laboratories in power electronics"; Proceedings of the IEEE Annual Power Electronics Specialists Conference, 430-436, 2008.

[46] M. G. Helander, M. R. Emami. "Engineering eLaboratories: Integration of remote access and eCollaboration"; International Journal of Engineering Education, vol. 24, n°3, 466-479, 2008.

[47] A. S. Patil. A. Abraham. "Intelligent and interactive web-based tutoring system in engineering education: Reviews, perspectives and development"; Studies in Computational Intelligence, vol. 273, 79-97, 2010.

[48] D. Hercog, B. Gergič, S. Uran, K. Jezernik. "A DSP-based remote control laboratory"; IEEE Transactions on Industrial Electronics, vol. 54, n°6, 3057-3068, 2007.

[49] P. Bauer, J. Dudak, D. Maga, V. Hajek. "Distance practical education for power electronics"; International Journal of Engineering Education, vol. 23, n°6, 1210-1218, 2007.

[50] C. Salzmann, D. Gillet, P. Huguenin. "Introduction to real-time control using LabVIEW™ with an application to distance learning"; International Journal of Engineering Education, vol. 16, n°3, 255-272, 2000.

[51] K. K. Tan, T. H. Lee, F. M. Leu. "Development of a distant laboratory using LabVIEW, International Journal of Engineering Education, vol. 16, n°3, pp. 273-282, 2000.

[52] S. Azaklar, H. Korkmaz. "A remotely accessible and configurable electronics laboratory implementation by using LabVIEW"; Computer Applications in Engineering Education, vol. 18, n°4, 709-720, 2010.

[53] K. Yeung, J. Huang. "Development of a remote-access laboratory: A DC motor control experiment"; Computers in Industry, vol. 52, n°3, 305-311, 2003.

[54] K. K. Tan, T. H. Lee, F. M. Leu. "Development of a distant laboratory using LabVIEW"; International Journal of Engineering Education, vol. 16, n°3, 273-282, 2000.

[55] C. N. Thai, B. L. Upchurch. "Tele-experimentation for machine vision course using netmeeting & LabVIEW software"; Computers in Education Journal, vol. 14, n°1, 2-11, 2004.

[56] (2009) INSAT nouveau programme ingénieur website. [Online].Available:http://www.insat.rnu.tn/images/pdf/programme-ingenieur_new-regime.pdf

Design of Graph based model for LQ based Adaptive Dynamic Courseware

RANJAN DASGUPTA[1] ,KAUSIK HALDER[2], NABENDU CHAKI[3]

1 Dept. of Computer Sc. & Engg., National Institute of Technical Teachers' Training & Research, Kolkata, India

2 Dept. of Computer Sc. & Engg., National Institute of Technical Teachers' Training & Research, Kolkata, India (student)

3 Dept. of Computer Science & Engineering, University of Calcutta, Kolkata, India

Abstract: *Major objective of the web-based learning is to design customized courseware taking into account of the input knowledge level and the specific objective of each individual learner. In order to improve the design of adaptive dynamic courseware to meet the requirement of individual learner we have already presented concept of Learner Quanta (LQ) & Learner Quanta Cloud (LQC) [9]. In this paper, concepts of the learner-centric, self-adaptive modular LQ have been extended through a Graph based model in order to identify the appropriate data structure of the Learner Quanta Cloud. To identify required LQ from the LQC and to test that whether a learner should able to meet his/her objective or not we have presented an algorithm based on LQ Graph. Subsequent matrix representation of the proposed graph helps us to define several properties. Reachability issue has been dealt in detail where other properties are also mentioned in this paper.*

Keywords*-Web-based learning, Learner Quanta, Learner Quanta Graph, Reachability*

1.0 Introduction

Adaptive Web-based educational systems emerged as an alternative to the traditional "one-size-fits-all" approach in the development of educational courseware. Adaptive Learner Quanta Based Dynamic Courseware [9] is a new direction of research within the area of adaptive web based learning. Adaptive dynamic courseware is building a model for the individual user and applies it for adaptation to that user.

In last decade, authoring and delivery of adaptable e-learning courseware appears to be very important for design of modern learning management platforms. During that period, there have been proposed a lot of works identifying the key challenges in adaptive web based information delivery. The main goal of personalized and adaptive e-learning was formulated in [14] as assuring of "e-learning content, activities and collaboration, adapted to the specific needs and influenced by specific preferences and context of the student". Some research groups focus on adaptability to learners' current knowledge based on the theory of knowledge spaces.

The identification of specific requirements for individual learners of vastly different background and the design of effective courseware for the use of distance learning are being considered as a challenging problem all over the globe. Several efforts [4, 6, 7] in this regard have been reported which are mostly course-specific. The non-linear way of storage of information in the form of hypertext has brought a revolutionary change in the teaching-learning process [1,2,3,5,7,12]. In the hypertext document, links have been established in such a way that the user can explore, browse and search for not only a particular item but can also get information regarding relevant/associated issues.

Cockertion and Shimell [4] evaluated hypermedia document as a learning tool. Vassileva and Deters [11] designed a dynamic courseware generator tool based on AI Planning techniques.

Keith S. Taber and his associates [10] at the University of Cambridge put forward a project aimed to integrate English and Science standards using technology as a vehicle.

The emphasis, however, was merely to improve the presentation of the learning material. Dr. John Munro [8] of the University of Melbourne has also worked on identification of the requirements for the effective delivery of course content. He essentially tried to analyze learning based on some pre-defined key issues. Guanon Zhang [13] has designed a computer based knowledge system for assisting persons in making decisions and predictions upon human or data-mining knowledge. Amongst recent works Shahin, Barakat, Mahmoud & Alkassar [18] tried to develop adaptive course which deals with previous knowledge and preferences. However it does not deal with any graph model. Other works like [19, 20, 21] deal with other various issues including courseware authoring, technology standards etc.

In this paper, we tried to represent Learner Quanta Cloud as a directed graph G (P, E) whose vertex (P) can be thought of as an LQ with collection of the set of input knowledge elements & output objective elements and edges (E) of two types. The first group is directed path from output objective element of one LQ to some input knowledge element of other LQ if exists in the LQ cloud. If exists, the first LQ can be considered as a pre-requisite LQ for the second. Of course, one LQ might have multiple pre-requisite LQs and in such case, edges from each such pre-requisite LQs to the LQ under consideration should be present in the graph and it can only be studied once all the pre-requisite LQs are studied.

Another type of edges exists which connects every input knowledge elements to every output objective element within an LQ. These are internal to the LQs and signify that all such input knowledge elements are required to study and once it studied properly (subject to evaluation not discussed here) the output objective elements are achieved. Formal definition of vertices & edges are given in article 3.0.

Once the graph is formed, we represent the graph by a suitable matrix and apply algorithms to analyze it. Various types of queries can thus be handled with the help of this matrix and at present we are working on the *reachability* issue which deals with the query which determines whether a user with certain background can be elevated to another level with the help of an LQC.

2.0 Related Work

A Learner's Quantum (LQ) of study is a measured part of a topic with a specific set of output objectives and requires a specific set of input knowledge on part of the learner. Our earlier works [9] [15] [16] present the LQ model where a course is sub-divided into several topics or LQ. A Learners' Quanta Cloud (LQC) is a collection of semantically related group of learners' quanta. Any arbitrary learner quanta could be part of more than one LQC, where LQs are grouped based on different semantics.

In [9] [15] [16] subset of LQC are selected which caters the requirement of any user having some background knowledge. An algorithm has been designed which ensures the proper learning sequence of the selected LQs so that the learner can navigate through it [9]. Besides, the pool of LQs is so designed that it would be adaptive to cater to a wide spectrum of learners with varying requirements.

In [17] we tried to present the proposed algorithm through UML representation, which elicits the methodology in a simple way. The modular and incremental architecture of the LQ based algorithm makes it an ideal analytical model for UML based analysis, design and development.

This paper is an extension of our earlier works [9] [15] [16] & [17] on design of adaptive dynamic courseware based on Learner Quanta and here we try to develop an appropriate graph model for Learner Quanta Cloud, which would help us in optimizing request of learners.

3.0 The Learner's Quanta Graph:

A Learners' Quanta Graph is a directed graph G= (P,E) whose vertex set P can be divided into two sets Source Vertex i.e. U= $\{u_1, u_2, ...u_n\}$ and Destination vertex i.e. V= $\{v_1, v_2, ...v_n\}$ and a set of edges E= $\{e_1, e_2 ....e_n\}$ such that every edge e_k connects a vertex in U to one in V or vice versa. Edges E are classified as inner edge and outer edge also.

The two sets U and V may be thought as two types of vertices: Source Vertex and Destination Vertex. Edges forming with a pair of vertices e= $(\overrightarrow{u_i, v_j})$ where direction is from u_i to v_j it is called Inner edge & edges forming with a pair of vertices e= $(\overleftarrow{u_i, v_j})$ where direction is from v_j to u_i it is called Outer edge

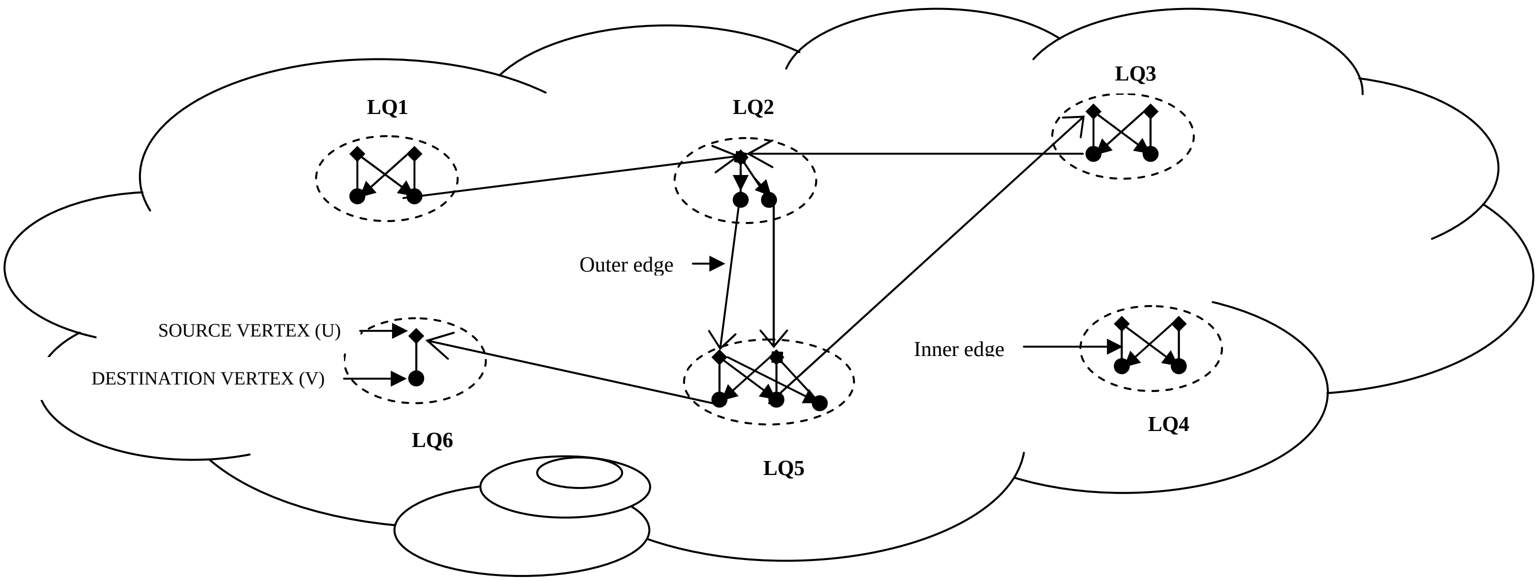


Fig: 1 Directed LQ Graph (G) in a LQ Cloud

Fig: 1 represents a directed Learner Quanta Graph in a LQ Cloud with 6 LQ (shown in dotted circle) which has 10 source vertices (diamond headed) & 12 destination vertices (round headed) and connected themselves through 21 inner edges and 6 outer edges

Each vertex of vertex set U & V is attached with any Learner Quanta (LQ) in Learner Quanta Cloud. We can also say that vertices of e i.e. u_i and v_j are always belongs to same Learner Quanta in case of Inner edge; never belongs to same Learner Quanta in case of Outer edge.

4.0 Matrix Representation of Learner Quanta Graph

Let G be a directed LQ graph where r is the total no. elements in U and s is the total no. elements in V. Then the LQ matrix $X = [X_{i,j}]$ of the LQ Graph is r by s (r, s) (0, 1,-1) - matrix whose elements

$x_{i,j}$ =1, if there is an edge directed from u_i to v_j

=-1 if there is an edge directed from v_j to u_i

=0 otherwise

Source vertex (u_i) or Destination vertex (v_j) can be represented numerically in the rows & columns of matrix with respect to a LQ as $k\alpha_m s_n$, Where k= either 0 or 1 (0 for source vertex & 1 for destination vertex respectively), α_m = {1, 2…n} implies LQ Number and s_n = {1, 2…n} implies Sequence Number of source vertex or destination vertex within a LQ

5.0 Benefits of matrix represent - tation of LQ Cloud/Graph

A lot of benefits can be achieved once the LQ cloud has been represented in LQ graph and subsequently to LQ matrix. In the following sections, we discuss a few of them. Reachability being on of the prime importance, it has been dealt in details, whereas the other issues are mentioned in brief

5.1 Pre-requisite, alternate pre-requisite and post-study LQ

In Learner Quanta Cloud when an output of LQ is matching (i.e. equivalent from academic knowledge perspective) with input of another LQ then first LQ is known as the *pre-requisite* of the second LQ and the second LQ is called the *post-study* LQ. Every LQ may have one or more pre-requisite LQs and also one or more post-study LQs.

Source Vertex (u_i) →

Destination Vertex (v_j) ↓

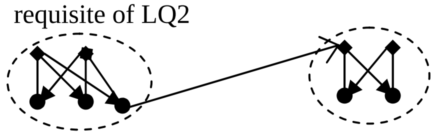
	011	**012**	**021**	**031**	**032**	**041**	**042**	**051**	**052**	**061**
111	1	1	0	0	0	0	0	0	0	0
112	1	1	-1	0	0	0	0	0	0	0
121	0	0	1	0	0	0	0	0	-1	0
122	0	0	1	0	0	0	0	-1	0	0
131	0	0	-1	1	1	0	0	0	0	0
132	0	0	0	1	1	0	0	0	0	0
141	0	0	0	0	0	1	1	0	0	0
142	0	0	0	0	0	1	1	0	0	0
151	0	0	0	0	0	0	0	1	1	0
152	0	0	0	-1	0	0	0	1	1	0
153	0	0	0	0	0	0	0	1	1	-1
161	0	0	0	0	0	0	0	0	0	1

.

Fig2: Matrix representation of directed LQ Graph (G) of Fig: 1

Study of all pre-requisite LQs is a necessary condition (except alternate pre-requisite LQs, if exists; defined later) before studying the post-study LQ unless learner has the same academic knowledge beforehand.

In fig: 3 one output of LQ1 is connected with one input of LQ2 showing that LQ1 is a pre-requisite of LQ2



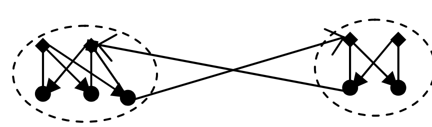
LQ1 **LQ2**

Fig3: LQ showing pre-requisiteness

This may be determined from the LQ matrix $X = [X_{i,j}]$ by checking v_j row entries equals to -1 for a particular u_i.
Alternate pre-requisite LQ: If input of one LQ is connected with output of more than one LQ, in that case all those LQs are called alternate pre-requisite LQs and for studying of the first LQ, any one of the alternate pre-requisite can be studied beforehand. For all those alternate pre-requisite LQs, u_i column entries will be -1.

5.2 Inconsistent state

Inconsistent state may occur when an output of a LQ matches with an input of another LQ and at the same time an output of the second LQ also matches with an input of the first LQ. These two LQ are pre-requisite of each other and so they can't be offered to the learner concurrently and they form an inconsistent state. Fig: 4 show a case of such inconsistent state:



LQ1 LQ2

Fig4: LQ showing Inconsistent State

This may be determined from the LQ matrix $X = [X_{i,j}]$ by checking row entries for two v_j belonging to two different LQ (LQ1 and LQ2 in fig.4) is -1 for two different u_i, where those two u_i must be a member of same LQ (LQ1 and LQ2) as v_j belongs. If above situation happens then it may be concluded that these two LQ are in inconsistent state.

5.3 Isolation

If there exists any LQ in LQC, whose input does not match with output of any other LQ and output does not match with input of any other LQ of the LQC then the said LQ remains in LQC as an isolated LQ and is called an isolated LQ. Isolated LQ can only be studied if user possesses input knowledge as required by the LQ and after study the learner cannot proceed further in the LQC.
This may be determined from the LQ matrix $X = [X_{i,j}]$ by checking occurrence of -1 in all row and column entries of a particular LQ. If no -1 available then we may say that it is an isolated LQ.

5.4 Reachability and its Test

Once the adjacency matrix as mentioned in article 4.0 is formed, we can easily find the reachability of the requirement (objective R) of learner having given his input knowledge (I). By the word 'reachability' we mean having given a learner input knowledge set I ,required output objective set R and a set of LQs(i.e. LQ Cloud), to find whether it is possible for the learner to elevate himself from level I to level R from the given LQ Cloud.
Article 5.4.1 shows the algorithm while article 5.4.2 illustrates step by step operation of the algorithm for two cases.

5.4.1 Algorithm: LQ search through LQ Graph

Statement: Test whether Requirements submitted by user is reachable by a Learner Quanta Cloud or not and if reachable, then identify the LQs from the LQ Cloud

Input:
Given a LQ Cloud represented by its LQ matrix $X = [X_{i,j}]$
Input Knowledge set provided by the user I= {i1, i2,……………..im}
Requirement set provided by user R= {r1, r2,……………..rn}
Output:
Whether the requirement set(R) is reachable from the input knowledge set(I)
If reachable then find the relevant Learner Quanta(LQ)(stored in LQ pool L)

Method: We start the algorithm by taking r_i where i=1, as the first requirement and also assumes that if a requirement is reachable then their relevant LQs will be stored into a LQ pool (L). The algorithm is as follows:

begin

Step1:
Check whether R is a proper subset of V; if not
Show "UNREACHABLE"
else
do Step2 to Step4
Step2:

For every element of R find the corresponding element in V such that

$$r_i = v_j$$

Step3:

For each v_j find the u_i for which there is a '1' in the adjacency matrix in the corresponding row and find respective u_i
Step4:

For every u_i check whether it is an element of I; if yes
Store the LQ no. in the LQ pool (L) and remove r_i from R;
Check
if R= ϕ then
{
Show "REACHABLE";
Print LQ Pool (L);
}
end
else do until i=n else
Find the '-1' in the corresponding column entries;

If no '-1' available then
{
Show "UNREACHABLE";
}
else
Find the corresponding v_j and for each v_j
Go to step 3;
do until i=n;
end

5.4.2 Illustrative Example

Example1: Let us assume that in first example we got one user whose requirement set is {112,121} and the user has Input Knowledge set as specified is {011,012,032,043}.We can now apply our algorithm on the matrix (fig:1) to check whether we can find its reachability.

Example2: Let us assume that in second example we got one user whose requirement set is {142,152} and the user has Input Knowledge set as specified is {041,051,052}. We can now apply our algorithm on the matrix (fig:1) to check whether we can find its reachability.

Solution:

The algorithm is again presented in the left when step by step operation for the above case shown in the right

Algorithm	Step by step operation of Example 1 of 5.4.2	Step by step operation of Example 2 of 5.4.2
Method: We start the algorithm by taking r_i where i=1…….n	**Method:** We start the algorithm by taking r_i where i=1….2	**Method:** We start the algorithm by taking r_i where i=1….2
Step1: Check whether R is a proper subset of V; if not Show "UNREACHABLE" else do Step2 to Step4	Check $R \subset V$ or not {112,121} $\subset$ {111,112,121,122, 131,132,141,142,151,152,153,161}; /* the answer being true, we go to step 2 to step 4*/	Check $R \subset V$ or not {142,152} $\subset$ {111,112,121, 122, 131,132,141,142,151,152,153,161}; /* the answer being true, we go to step 2 to step 4*/
Step2: For every element of R find the corresponding element in V such that $r_i = v_j$	Since r_i ={112,121} corresponding v_j ={112,121}	Since r_i ={142,152} corresponding v_j ={142,152}
Step 3: For each v_j find the u_i for which there is a '1' in the adjacency matrix in the corresponding row and find respective u_i	For v_j =112 required u_i ={011,012} For v_j =121 required u_i ={021}	For v_j =142 required u_i ={041,042} For v_j =152 required u_i ={051,052,053}
Step4: For every u_i check whether it is an element of I; if yes Store the LQ no. in the LQ pool (L) and remove r_i from R; Check	For v_j =112 Check {011,012} $\subset$ {011,012,032,043} /* the answer being true*/ Store LQ no=1 to LQ Pool(L) R= {112,121}-{112}={121} Check	For v_j =142 Check {041,042} $\not\subset$ {041,051,052} /* the answer being false*/ For 042 No -1 available in the corresponding column Show "UNREACHABLE"

if R=ϕ then
{
Show REACHABLE";
Print LQ Pool (L);
}end
else do until i=n
else
Find the '-1' in the corresponding column entries;
If no '-1' available then
{
Show "UNREACHABLE";
}
ElseFind the corresponding v_j and for each v_j
Go to step 3;
do until i=n; end
	$\{121\} \neq \phi$
	/* the answer being false we are going to next element of R */
	For v_j =121
	Check
	$\{021\} \not\subset \{011,012,032,043\}$
	v_j ={112}/* for which -1 in the column of 021 */
	For v_j =112 required
	u_i ={011,012} and $u_i \subset$ I
	Store the LQ no.=2 to the LQ Pool(L)
	R= {121}-{121}=ϕ
	Show "REACHABLE"
	Print 1,2
		end

6. Conclusions

Learner Quanta based Adaptive Dynamic Courseware generation continues as a challenging issue for modern e-learning. The paper aimed at presenting an appropriate structure for the adaptable content within the context of Learner Quanta Cloud [9]. The concept of LQ model [9] has been augmented in this paper by introducing LQ Graph & subsequently LQ Matrix. Both the LQ Graph and LQ matrix described over here are specially designed to support adaptability. This LQ matrix in turn helps a lot in identifying various issues related to management of LQs. Among the key benefits we offer from the LQ Graph are Test of isolation, reachability, identification of appropriate LQ and others have been discussed and more such benefits are expected to be available from the study of LQ matrix.

7. References

[1] Abdel H., et. al. : Multimedia Integration into a Distance Learning Environment: Proc. of the 3rd International Conference on Multimedia Modeling, Tolouse. (1996)

[2] Braun T. : Multimedia Application in Networks in Multimedia Communications – Protocols and Applications edited by Kuo. Franklin, Wolfgang Effelsberg, J. J. Garcia-Luna-Aceves, Prentice Hall PTR (1998).

[3] Chang C. K., Chen G. D. : Constructing collaborative learning activities for distance CAL systems: Journal of Computer Assisted Learning, Vol. 13, No. 1 (1997).

[4]Cockerton T., Shimell R. : Evaluation of a hypermedia document as a learning tool : Journal of Computer Assisted Learning, Vol. 13, No. 2 (1997).

[5] Crook C. K. : Making hypertext lecture notes more interactive: undergraduate reaction in the Journal of Computer Assisted Learning, Vol. 13, No. 4 (1997).

[6] IEEE LTSC: Learning Technology Systems Architecture (LTSA), Version 4.00. URL: http://www.edutool.com/ltsa (1998).

[7] Kuo F. : Introduction to Multimedia in Multimedia Communications – Protocols and Applications edited by Kuo Franklin, Wolfgang Effelsberg, J. J. Garcia-Luna-Aceves – Prentice Hall PTR (1998).

[8]Munro John: Facilitating Effective Learning and Teaching: Proc. of Technology Colleges Trust Online Conference (2002). URL:www.cybertext.net.au/tct2002/keynote/printable/munro.htm

[9]Ray S., Chaki N., Dasgupta R. : Design of an adaptive web-based courseware : Proc. of IASTED International Conference on Intelligent Systems & Control (ISC 2004), Honolulu, Hawaii, USA, (2004).

[10]Taber K. S. : Development of Student Understanding: A Case Study of Stability and Lability in Cognitive Structure, Research in Science & Technological Education, Vol 13 (1), (1995) pp.87-97.

[11] Vassileva J., Deters R. : Dynamic Courseware Generation on the WWW : British Journal of Educational Technology, Vol. 29, No.1 (1998).

[12] Wiles N., Wright D. : A hypermedia system to aid teacher understanding a pupil preconceptions: Journal of Computer Assisted Learning, Vol. 13, No. 1, (1997).

[13] Zhang George Guanon : Computer Based Knowledge System in the USPTO United States Patent Office, September, 2004. URL: http://www.uspto.gov/Patents/United States Patent 6,795,815.htm

[14]Dagger, D., Wade, V. & Conlan, O., 2005, Personalization for All: Making Adaptive Course Composition Easy. Special issue of the Educational Technology and Society journal, IEEE IFETS.

[15] S. Ray, R Dasgupta, N Chaki; "A Learner's Quant Model based Framework towards Building Dynamic Web-based Courseware"; Proc. of the fourth International Conference on Multimedia and ICTs in Education m-ICTE2006) - "Current Developments in Technology-Assisted Education", Seville, Spain, pp. 238-242, November 2006

[16]N. Chaki, R. Dasgupta, "A Learners' Quanta Based Framework for Identification of Requirements and Automated Design of Dynamic Web-based Courseware" Invited Talk, 14th International Monterey Workshop, Monterey, California, USA, September, 2007.

[17]Halder K, Pal S, Chaki N , Dasgupta R " UML modeling of Adaptive course ware algorithm" 2008 in World Science Congress on Computer Science & Engineering-2008 San Francisco, USA (ICEIT-08) 2008

[18] Reem Shahin, Lina Barakat, Samhar Mahmoud, Mohammad Alkassar."Dynamic Generation of Adaptive Courses" Information and Communication Technologies: From theory to applications2008, IEEE 2008

[19]C. Hong,C.Chen,M Chang,S Chen: Intelligent Web-based Tutoring System with personalized Learning Path Guidance-7th IEEE International Conference of Advance Learning Technologies(ICALT 2007)

[20] B.Bontechv,D Vassileva: Courseware authoring for adaptive e-learning-International Conference on Educational Technology and Computer-2009 DOI10.1109/ICETC.2009.59

[21] J Zhang : Dynamic Generation of the Adaptive Courseware Based on Learning Technology Standards- Proc. Of the 5th International Conference on Computer Science & Education, Hefei China August 2010 pp.741-745

A Framework for Implementing Virtual and Remote Laboratories in Scientific Course

L. de la Torre[1], R. Heradio[1], H. Vargas[2], J. Sanchez[1], and S. Dormido[1]
[1]Computer Science and Automatic Control, Spanish Open University (UNED), Madrid, Spain
[2]School of Electrical Engineering, Univ. Catolica de Valparaiso, Valparaiso, Chile

Abstract— *This work describes the experience of the authors regarding the design, development, analysis, and exploitation of web-based technologies for creating an online experimentation framework with distance education purposes. The framework we propose is specially designed for scientific and technical courses that require a quite important presence of experimentation. It supports the development of virtual and remote laboratories, and their integration in a Learning Management System. In addition, to illustrate the applicability of our proposal this paper presents a network of remote and virtual laboratories for control engineering education that has been implemented with the framework.*

Keywords: Remote labs, Virtual labs, Distance education, Control engineering education, Web-based technologies.

1. Introduction

It is commonly accepted that digital media (such as simulations, videos, interactive screen experiments or web laboratories) can positively impact students knowledge, skills and attitudes [1]. Consequently, tools such as Learning Management Systems (LMSs) and web–based laboratories have become widespread in distance education in the last decade. LMSs support the administration, documentation, tracking, and reporting of training programs, classroom and online events [2]. Web–based laboratories make possible to illustrate scientific phenomena that require costly or difficult-to-assemble equipment [3], [4].

There are two complementary approaches for web–based laboratories:

1) Virtual Laboratories (VLs) provide computer based simulations which offer similar views and ways of work to their traditional counterparts. Nowadays, simulations have evolved into interactive Graphical User Interfaces (GUIs) [5] where students can manipulate the experiment parameters and explore its evolution.
2) Remote Laboratories (RLs) use real plants and physical devices which are teleoperated in real time [6].

As far as we know, proposals on the development of VLs and RLs are ad hoc solutions to particular problems [7], [8] that rarely support a social context for the interaction and collaboration among students (and between teachers and students) [9], [10]. Such context can be supported by current LMSs. Nevertheless, creating VLs and RLs from scratch, and integrating them into LMSs involve a huge effort. In order to minimize such effort, this paper proposes a conceptual framework to cost–effectively build VLs and RLs that are deployed with LMSs.

Our framework is based on the following software tools:

1) The LMS Moodle[1] to create collaborative web learning environments.
2) The java code generator Easy Java Simulations (EJS)[2] to easily develop Java applets that serve as GUIs in both VLs and RLs.
3) The graphical development environment LabView[3] to write programs to control remote hardware in RLs.

The remainder of the paper is organized as follows: Section 2 provides a global vision of our framework, its architecture, and the tools it uses. Section 3 presents an instantiation example of the framework applied to a real control engineering course that is being taught in Spanish University for Distance Education (UNED). In Section 4 students' assessments are presented and analyzed while we also discuss the advantages and need of these kind of portals within the distance education context. Finally, in Section 5 some conclusions and future work lines are given.

2. E-learning Platform Framework

This section presents our general framework for implementing VLs and RLs, and deploying them into the LMS Moodle. Figure 1 shows the UML deployment diagram of the framework, which is composed of the following nodes:

1) The *E–learning Server* node includes two components which form the e-learning resources: the *LMS* and the *Booking System*. The LMS provides students with all the theoretical documentation, protocol tasks, and complementary information they may need as well as communication channels between students and professors. The booking system component is in charge of scheduling the access to the hardware resources of the remote laboratories.

[1]Moodle website: `http://moodle.org/`
[2]EJS website: `http://www.um.es/fem/Ejs/`
[3]LabView website: `http://www.ni.com/labview/`

2) The *Client* node includes the *Virtual/Remote Laboratory* component. It implements a VL, a RL, or both of them. It is developed with EJS as a Java applet.
3) The *Physical Laboratory Server* node includes *the Controller of Physical Devices* component. It is implemented with LabView and is responsible for controlling the devices of the real laboratory (e.g., actuators, sensors, cameras...).

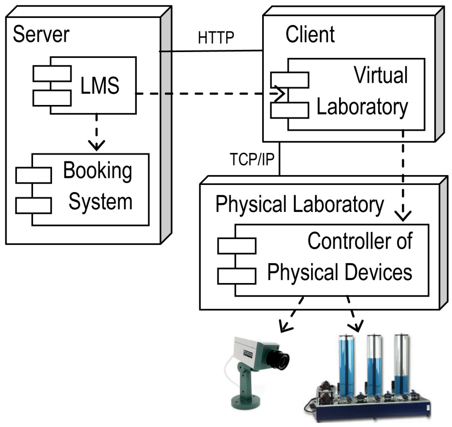


Fig. 1: Deployment diagram (UML).

Section 2.1 describes the implementation of the nodes *Client* and *Physical Laboratory Server*. Section 2.2 sums up the implementation of the node *E-learning Server*.

2.1 Implementation of Experimentation Resources

In our framework, each laboratory deployed in the e-learning platform is implemented by a pair of components of type *Virtual/Remote Laboratory* and *Controller of Physical Devices*.

2.1.1 Virtual/Remote Laboratory

The GUI applications are distributed as Java applets (developed with EJS) to Web browsers. This is represented by the HTTP communication link between the server and the client nodes of Figure 1. By means of the applet GUI, students can observe the effect in the dynamic behavior of the processes during their virtual or remote manipulation.

During the -virtual or remote- experimental sessions, students can also save data registers (parameters and measurements) or images (graphs with plotted data) of the experiments for later analysis. These files are saved in the server, sent from the applet via FTP using the Linlyn class[4] and get stored in the personal folder corresponding to the students Moodle account. This way, students can access them from any computer with an Internet connection and use them to write the laboratory reports which are finally sent to the instructors by means of a submission application of this LMS.

2.1.2 Physical Laboratory Application

The teleoperation of the physical devices in the real laboratory is addressed by means of a client and server architecture. The TCP/IP communication link between the client and the physical laboratory nodes and the dependency lines between the hardware and the controller of physical devices component of Figure 1 show the basic structure of these applications, where a remote client manipulates processes located in the laboratory through a server computer working as a middleware communication layer. Visual feedback of the distant equipment is usually provided by webcams that point to the real equipment.

TCP/UDP links are commonly used for exchanging data and commands between both sides based on a design pattern known as command-based architecture. The laboratory side executes three tasks concurrently: the Command Parser, the Sender, and the Acquisition and Control-Loop. The command parser receives commands from the client, interprets them, and executes the requested actions. When no request is received, the command stays idle, leaving the processor free for other duties. Similarly, the sender "sends" to the client the measurements acquired by the control loop when a command requires it. The acquisition and control-loop thread performs the data acquisition and closed-loop control of the process. On the other side, the client application also implements the transmission layer needed to exchange data with the physical laboratory (Sender and Receiver threads). A third task is the rendering of the information to the final users.

This framework uses a novelty approach to make the creation process of the experimentation resources in an easier way, which relies on the use of two software tools especially adequate for developing these experimentation resources: LabView and EJS. The approach is based on the creation of generic communication modules both in the client and the laboratory sides. On the client-side, a Java library called *jil* with a generic communication interface has been created. By this class the TCP protocol is hidden to users and simple Java classes/methods are provided to set up the connections. This library can be easily integrated into EJS programs to dialog with the server. Similarly, on the laboratory-side, a LabVIEW executable program called JiL Server operates as a middleware communication layer between the client and the plant. Thus, developers are only required to create a local control Virtual Instrument (VI) in LabView that performs the acquisition and the closed-loop control of the plant. More information about this approach can be found in [11].

2.2 Implementation of E-learning Resources

In spite of the advances in computer based simulations (and in remote real experimentation), few works have been

[4] http://www.uwplatt.edu/csse/courses/prev/s02/se273/comm/Linlyn.java

made to study the impact of different implementation methods of digital distance learning objects, which is a fundamental topic that teachers have to confront [12]. VLs and RLs do not provide by themselves all the convenient resources for distance teaching/learning of students with all the implications that this methodology involves. Specifically, students must carry out their practical activities in an autonomous way and therefore, complementary Web based resources to the virtual and remote labs should be included. For this reason, for each VL/RL there should be available, not only a description of the phenomena under study and of the didactic setup of the experiment for remote experimentation, but also the task protocol students must follow to achieve the proposed goals. Moreover, a laboratory report must be prepared by the students with the data collected during the simulated and real experimentation that the instructors will correct. Thus, a second key aspect to be addressed is the development of a web platform that offers to students a personal online workspace and supports their learning process with the previous (and others) resources.

This e-learning platform organizes the access of users to available experimentation modules and simplifies the organization of user groups. It should also be able to offer notification services by email, instant messaging inside the online portal, news, forums, etc. allowing the interaction and the collaboration among students (and teachers/students).

The website should provide all the necessary theoretical documentation like practical guides, tasks protocols, instructions manuals or any other kind of information needed to satisfactorily perform a remote experimentation session in an autonomous way.

The LMS could also suggest or impose a sequence of tasks or activities that students should carry out during an experimental session. The tasks can be of two types: Firstly, the tasks which students must carry out before performing the experiments in the real plant. This work should be done with a GUI that allows students to work in simulation mode. The aim of this first step is to get an adequate previous insight about the process. This way, students will reduce the time spent in the activities that work over the real plant. The access in remote mode should not be allowed until the student has completed the tasks in simulation mode. Once the student's work in simulation mode has been evaluated by the teaching staff, the access in remote mode can be granted.

Another important task that must be taken into account is the management of students and their assessments as well as the uploading of reports and the tracking of them. Also, a personal online file repository for students where the data collected during their experimentation sessions are stored.

Finally, an automatic booking system must be included in the e-learning platform to schedule the access to the physical resources of the laboratory for the RLs can only be used by one person at the same time. Therefore, a special application must be created to take care about the scheduling of these hardware resources. Some LMSs also include booking systems that can be adapted and used for this purpose.

At the end of the development process, both kinds of resources are integrated to produce the Web environment. Moodle, a free LMS e-learning platform that offers all the necessary tools to cover the implementation requirements mentioned above as well as some additional interesting features, is used in this framework.

3. A Framework Instantiation Example: AutomatLabs

This section presents an example of application of the framework previously detailed to a university course about automatic and control.

The Department of Computer Science and Automatic Control of the UNED has been using VLs and RLs in the last years to give response to the demand from students who have difficulties to attend classes in the academic centers. By means of these tools, students are able to have a first direct contact with the following systems: coupled three-tanks system, a temperature control system and a direct current servo-motor. The previous virtual and remote control laboratories can be appreciated in the left part of Figure 2. The upper picture shows the three-tanks system, a MIMO system where liquid level control experiments can be carried out [13]. Multivariable control concepts can be studied and put into practice using this laboratory. The one in the middle depicts the Heat-flow system that allows performing practical experiments on systems with transport delays. Finally, the bottom one shows the DC Motor, a SISO system that allows studying the dynamic behavior in speed and position of a motor fed by a direct current source [14].

Every lab has two working modes: virtual (based on a mathematical model of the process) and remote (which access to the real plant). The GUIs are divided into two parts. The left part contains a graphical representation of the plant and a control panel used to define different system parameters. The virtual representation has been developed by copying the actual hardware. Thus, any variation of the system state during the simulation will be automatically represented over the virtual scheme. On the other hand, when a user works in remote mode, this virtual representation is replaced by video images sent from the server. In this working mode, an augmented reality option is offered (Heat-flow remote lab in Figure 2) and so the virtual representation of the process can be overlapped with the video image.

Figure 2 also shows the main page of the AutomatLabs portal, listing all the 3 experimental blocks integrated into the Moodle portal. Each experimental block provides all the resources students may need to perform the online experimentation by themselves. Such resources are structured as follows: first link (blue line) opens the related

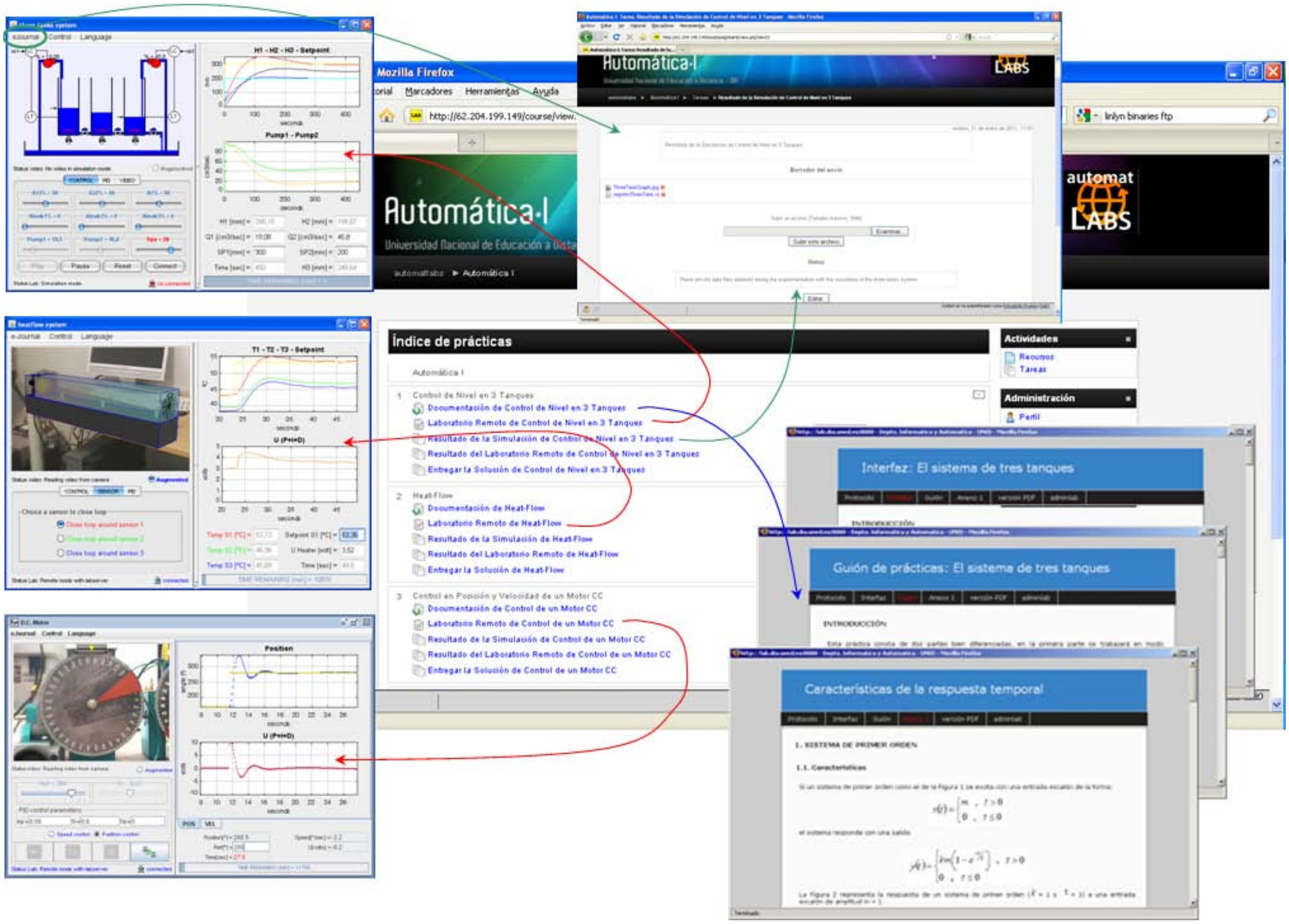


Fig. 2: AutomatLabs web page and its online resources.

documentation, protocol tasks, etc.; second link (red lines) opens the corresponding Java applet with the VL and RL (shown at the left of that same figure); third and forth links direct towards the file repositories where the data collected during the virtual and remote experimentation are stored, respectively; and finally, the last link from each experimental block opens the submission application to let students send their laboratory report to the instructors.

AutomatLabs uses a custom made booking system but there are free Moodle blocks such as RMBS[5] that can be used instead. Moodle allows interaction among students (and between students and professors) by means of instant messages or forums (as long as the instructors have enabled these resources). Theoretical tests can be defined to evaluate students knowledge after reading the provided documentation and before allowing his access to the experimentation resources (the EJS Java applets). Instructors can also use Moodle to fix deadlines to perform the previous activities and these events become visible in calendars inside the e-learning platform for all the students.

[5] http://moodle.org/mod/data/view.php?d=13&rid=734

4. System Assessment and Relevance

4.1 Students' Assessment

AutomatLabs was tested by 20 students of the UNED during their third grade year on Computer Sciences in the Control Engineering subject. They were granted access to the three laboratories presented in section 3 for four months. After they finished their experimentation with this system, they were asked to fulfil an opinion poll with different questions about their experience using AutomatLabs. In this opinion poll, they were asked to mark several statements with numbers from 1 to 5, being 1 that they completely disagree with the statement and 5 that they completely agree with it. Four of the more important questions and answers are shown in Figure 3.

Where Qi are:

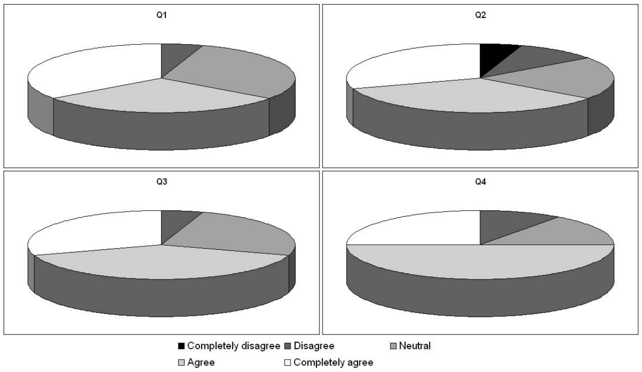


Fig. 3: Students answers to four questions about their perception of AutomatLabs.

1) Q1: Overall, I am satisfied with the clarity and easy of use of the system and environment used for the experimentation.
2) Q2: I was able to resolve all the tasks asked in the documentation just using the material given in this system.
3) Q3: Simulations (virtual laboratories) have served to make clearer how to work with the remote laboratories.
4) Q4: Overall, I am satisfied with this way of performing the experimental activities.

Q1 presents good results with just 5% of students that completely disagree or disagree and with 65% of positive answers. This means the integration of the resources into Moodle are natural for the students as they can easily use them. Q2 shows that almost all the students were able to resolve the experimentation tasks without any other help than the one given by the material provided in AutomatLabs which points out that our framework is able to offer and contain everything a student may need for these kind of courses. Q3 reveals the importance of simulations as a first experimental contact for the students in order to make clearer the use of the remote laboratory for 70% of the students agreed or completely agreed this statement and only 5% of them disagreed. This is not a necessary step in the framework we present but, as seen from the students' responses, it is nice to include it. Finally, Q4 may serve as a conclusion of the whole opinion poll: 75% of the students agreed or completely agreed that AutomatLabs satisfied them as a way to perform their experimental activities.

4.2 Relevance for Instructors

Traditional laboratories may sometimes present a low ratio between their use and their costs. By allowing a remote access to these laboratories, their frequency of use can be increased thanks to the creation of networks of educational institutions interested in the same shared experiments. Although the economical aspect is important, there are other benefits tied to these laboratories. The most obvious one is their improved accessibility with respect to their traditional counterparts: since they can be used from students' homes, anybody can access to them, even handicapped people. Another advantage of remote laboratories is their increased availability thanks to their capability to operate 24 hours a day without constant supervision. Finally, the safety of the laboratory devices is also guaranteed due to the software-controlled remote use of these resources.

5. Conclusions

In this work, we have presented a general framework which can be applied to any technical or scientific course that needs to provide experimentation as a complement to the theoretical lessons and offered an example of its implementation with AutomatLabs, a network of virtual and remote laboratories.

The proposed framework uses Java applets made with EJS, LabView to control the remote hardware, and a free LMS (Moodle) that provides and/or supports all the resources students may need: the virtual and remote laboratories, the documentation, the protocol tasks, the repository to store the files with the data obtained from the virtual or real experimentation, an application to submit the laboratory reports prepared by the students, a booking system, and communication channels between students and instructors as forums and instant messaging.

The VLs and RLs provide students with the necessary tools to experiment with different systems and put into practice the theoretical lessons learnt from the documentation. This can be easily done in an autonomous way following the tasks protocol prepared by the instructors for each laboratory. Reports with the images or numeric data obtained during the experimentation sessions are sent via FTP to the server and get stored separately for the different laboratories in each student's file repositories. These repositories allow students to download or delete these files as well as to make personal annotations or comments related to them.

A booking system is integrated in the e-learning platform to manage the scheduling of the hardware used by the RLs. Finally, communication channels such as forums and instant messaging are easily added thanks to Moodle. The final result is a framework that allows to create online portals where scientific or technical students can perform their experimentation sessions at distance and in an autonomous way.

AutomatLabs is an example of application of this framework to a university course about automatic control and offers to students three virtual and remote laboratories along with all the other resources and tools described above.

In order to facilitate the installation and distribution of our framework, we plan to improve its integration in Moodle, packing part of the framework as a Moodle module. This work will involve (1) avoiding the necessity of a FTP server to send the files with the collected data or images and (2) removing the use of a custom made booking system. The first goal can be achieved creating a Java library in charge of the communication with Moodle's files submission application. The second one could be achieved making some changes to one of the previously mentioned booking system Moodle blocks.

References

[1] R. Kozma, "Will media influence learning? reframing the debate," *Education Technology Research and Development*, vol. 42, no. 2, pp. 7–19, 1994.

[2] R. K. Ellis, "Field guide to learning management systems," tech. rep., ASTD Learning Circuits, 2009.

[3] G.-W. Chang, Z.-M. Yeh, H.-M. Chang, and S.-Y. Pan, "Teaching photonics laboratory using remote-control web technologies," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 642 – 651, 2005.

[4] S. Sivakumar, W. Robertson, M. Artimy, and N. Aslam, "A web-based remote interactive laboratory for internetworking education," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 586–598, 2005.

[5] J. Gratch, J. Kelly, and C. Bradley, "Science simulations: What do they contribute to student learning?," in *Society for Information Technology and Teacher Education International Conference*, (Chesapeake, USA), pp. 3422–3425 Chesapeake, USA, 2007.

[6] C. Salzmann and D. Gillet, "Challenges in remote laboratory sustainability," in *International Conference on Engineering Education.*, (Coimbra, Portugal), 2007.

[7] S.-T. Park, H. Lee, K.-C. Yuk, and H. Lee, "Web–based nuclear physics laboratory," in *Recent Research Developments in Learning Technologies*, pp. 1165–1169, 2005.

[8] F. Schauer, F. Lustig, and M. Ozvoldova, "Remote scientific experiments across internet: Photovoltaic cell characterization," in *International Conference on Interactive Collaborative Learning*, 2006.

[9] D. Gurkan, A. Mickelson, and D. Benhaddou, "Remote laboratories for optical circuits," *Education, IEEE Transactions on*, vol. 51, no. 1, pp. 53–60, 2008.

[10] M. Macias and I. Mendez, "elab - remote electronics lab in real time," in *Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual*, pp. S3G–12 –S3G–17, 2007.

[11] H. Vargas, J. Sanchez, C. Salzmann, F. Esquembre, D. Gillet, and S. Dormido, "Web-enabled remote scientific environments," *Computing in Science and Engineering*, vol. 11, pp. 34–36, 2009.

[12] H. Urban-Woldron, "Innovative teaching and learning scenarios using interactive simulations," in *Multimedia Physics Teaching and Learning*, (Udine, Italy), 2009.

[13] S. Dormido, H. Vargas, J. Sanchez, R. Dormido, N. Duro, S. Dormido-Canto, and F. Morilla, "Developing and implementing virtual and remote labs for control education: The uned pilot experience," in *17th IFAC World Congress*, 2008.

[14] C. Salzmann, D. Gillet, and P. Huguenin, "Introduction to real-time control using labview with an application to distance learning," *International Journal of Engineering Education*, vol. 16, pp. 255–272, 2000.

Transformations and Bases in OpenGL and the Viewing Pipeline

Hassan Farhat
Computer Science
University of Nebraska at Omaha

Abstract- *Geometric transformations are fundamental aspects of all computer graphics rendering and are found throughout the computer graphics pipeline. In earlier work we presented mathematical treatment of this topic. The different concepts were combined into a general mathematical procedure based on the concept of vector spaces in linear algebra.*

While we established the mathematical aspects of transformations, we found the application of this in the OpenGL environment was not simple, especially when the modelview matrix was constructed using the viewing matrix obtained by the "gluLookAt" function. When the gluLookAt function is used, incremental changes in viewing were possible by first re-computing the viewing matrix in world coordinates and not by composing with incremental changes in the viewing coordinates.

In this paper we look at alternative methods of application of the mathematical work in OpenGL. The two alternative procedures of rendering (transformations in world and transformation in viewing) are explored in the OpenGL environment.

1 Introduction

An essential aspect of computer graphics is the concept of transformations [6, 7]. Transformations are found through out the graphics pipeline (Figure 1). The figure shows a simplified view of the graphics pipeline in OpenGL [1, 2, 4].

Initially, objects are defined in their own *modeling coordinates* (example a circle with center at the origin and a unit radius). Instances of objects are then created in world coordinates by proper translation, rotation and scaling. Each instance with corresponding transformation corresponds to a copy of the object, properly placed in world coordinates.

In OpenGL, one invokes the MODELVIEW matrix to make copies of objects and place them in world or object coordinates [4, 5].

As the name implies, the model-view matrix can also be used to build viewing coordinates. The viewing coordinates are described by a set of orthonormal vectors and viewing origin. The coordinates of the vectors and the origin are given in world coordinates. In OpenGL, one can build the viewing coordinates using a call to the OpenGL graphics utility library (gluLookAt()).

In [3], we established the mathematical concepts that allowed the computation of coordinates of a point measured in several different coordinates system. When the concepts were applied to OpenGL using gluLookAt for example, we found incremental changes in viewing were not simple. In this paper, we look at how the transformation equations can be used effectively in OpenGL with or without the need to use the gluLookAt function.

The paper is organized as follows. In section 2, we review the computations of transformations needed to describe the coordinates of object in several coordinates system.

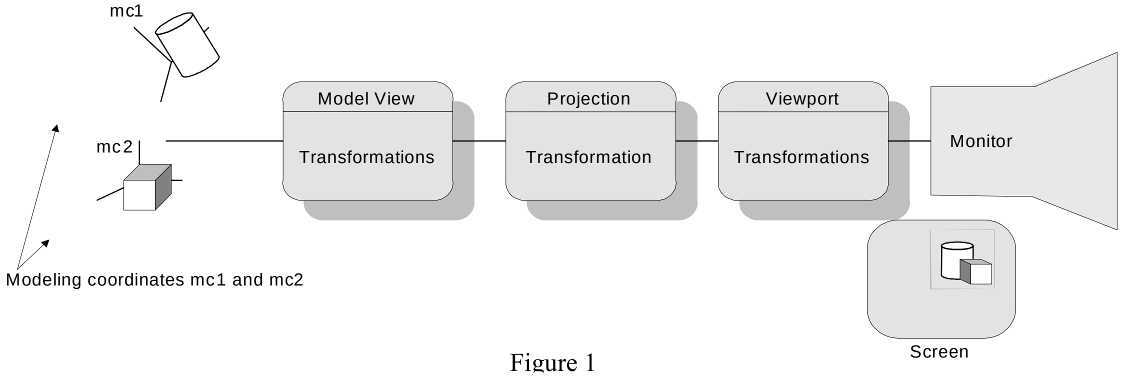


Figure 1

In section 3, we look at the application of these concepts in OpenGL when the gluLookAt function is used. In section 4 we look at alternative procedures where incremental changes in viewing can be applied. The conclusions are given in section 5.

2 Graphics Pipeline and Coordinates Transformations

As suggested by Figure 1, an object can be represented in several coordinate systems including: modeling, world, viewing and device coordinates as examples. Affine transformations are important in determining the coordinates of an object in several coordinate systems. They are also important in transforming an object within the same coordinate system.

Affine transformations for a three-dimensional space for points are described in homogeneous coordinates and are given by the equation

$$\begin{pmatrix} x_{new} \\ y_{new} \\ z_{new} \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = AX$$

In above, the a_{ij} values determine the type of transformation applied on the homogeneous point $(x, y, z, 1)^T$ [1]. The point $(x_{new}, x_{new}, x_{new}, 1)$ is the transformed point under A. The equation can also serve as a transformation of coordinate systems as follows. The point (x, y, z, 1) can be the coordinate of a point in space with respect to one coordinate system while the same point in space will have the coordinates $(x_{new}, y_{new}, z_{new}, 1)$ in another coordinate system.

When the equation is used as a transformation of coordinate system, we use the concept of a vector space. Even though points and vectors in n-dimensional coordinate system are described using n-tupple. The two differ in the type of operations applied to them. In [5] the distinction is made by associating points with the origin of a given coordinate frame.

A vector, X, written as a linear combination of other elements of a set of vectors, S = {X1, X2, …, Xn}, satisfies the equation

$$X = c_1 X1 + c_2 X2 + \ldots + c_n Xn \qquad (1)$$

for proper choice of scalars $c_1, c_2, .., c_n$. The set S is linearly independent if no vector in S can be written as a linear combination of other vectors in the set. A set, S, forms a basis for a vector space, V, if: a) every vector, X, in the vector space V can be written as a linear combination of vectors in S, and b) the set S is linearly independent.

For a set, S, forming a basis of a vector space, the elements of S constitute an alternative coordinate system to the traditional natural basis set. A vector, X, given in the natural basis have coordinates $(c_1, c_2, \ldots, c_n)$ in the coordinate system formed from the elements of S.

The coordinate transformation matrix: The coordinate transformation matrix for a general n-dimensional vector space, $\mathcal{V}$, and a subset basis of $\mathcal{V}$, S = {X1, X2, …., Xn} can be generated from the equation

X = (x1, x2, …, xn) = c1 X1 + c2 X2 + … + cn Xn

$$\begin{pmatrix} x1 \\ x2 \\ \vdots \\ xn \end{pmatrix} = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{nn} \end{pmatrix} . \begin{pmatrix} c1 \\ c2 \\ \vdots \\ cn \end{pmatrix} \qquad (2)$$

Equation (2) serves as a method of moving from one coordinate system to the other. When the Xi vectors are of unit length and are pair-wise perpendicular; i.e., when the bases are orthonormal, the ci terms can be computed as

ci = X . Xi

where the "." is the dot product. The above is then used to compute the needed transformation matrix used to write the coordinates of a vector in two different frames, label as j and j+1, as follows [3].

Given an orthonormal basis of a vector space {X1, X2, …, Xn} and an arbitrary vector X = {x1, x2, …, xn} with S and X given in coordinate frame j. The coordinates of X in frame (j + 1) with basis S (natural bases S) is

$$\begin{pmatrix} c1 \\ c2 \\ \vdots \\ cn \end{pmatrix} = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{nn} \end{pmatrix}^T . \begin{pmatrix} x1 \\ x2 \\ \vdots \\ xn \end{pmatrix} = M.X \qquad (3)$$

Note that due to the orthonormal property, the inverse of the matrix is its transpose.

Using this property, the above equation can be used to transform from one system to another in either direction (world to view or vice versa) without the need to compute the inverse of M. If the position vector C is given in viewing instead of X we obtain X as

[1] The T is the transpose of the column matrix representation of the point. We omit the T for the remaining points.

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & x_{2n}\\ x_{31} & x_{32} & \cdots & x_{3n}\\ \vdots & \vdots & \cdots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix}^{-1} \cdot \begin{pmatrix} c_1\\ c_2\\ c_3\\ \vdots\\ c_n \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & x_{2n}\\ x_{31} & x_{32} & \cdots & x_{3n}\\ \vdots & \vdots & \cdots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix}^{-1} \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & x_{2n}\\ x_{31} & x_{32} & \cdots & x_{3n}\\ \vdots & \vdots & \cdots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1\\ x_2\\ x_3\\ \vdots\\ x_n \end{pmatrix} \qquad (4)$$

Or

$$\begin{pmatrix} x_1\\ x_2\\ x_3\\ \vdots\\ x_n \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & x_{2n}\\ x_{31} & x_{32} & \cdots & x_{3n}\\ \vdots & \vdots & \cdots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix}^{T} \cdot \begin{pmatrix} c_1\\ c_2\\ c_3\\ \vdots\\ c_n \end{pmatrix} = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1}\\ x_{12} & x_{22} & \cdots & x_{n2}\\ x_{13} & x_{23} & \cdots & x_{n3}\\ \vdots & \vdots & \cdots & \vdots\\ x_{1n} & x_{2n} & \cdots & x_{nn} \end{pmatrix} \cdot \begin{pmatrix} c_1\\ c_2\\ c_3\\ \vdots\\ c_n \end{pmatrix} \qquad (5)$$

For the case where the coordinate systems origins do not match we obtain the equation [3]

$$\begin{pmatrix} c_1\\ c_2\\ \vdots\\ c_n\\ 1 \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} & 0\\ x_{21} & x_{22} & \cdots & x_{2n} & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nn} & 0\\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 & -P_{01}\\ 0 & 1 & \cdots & 0 & -P_{02}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \cdots & 1 & -P_{0n}\\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1\\ x_2\\ x_3\\ \vdots\\ x_n\\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} & -\sum_{i=1}^{n} x_{1i}.P_{0i}\\ x_{21} & x_{22} & \cdots & x_{2n} & \sum_{i=1}^{n} x_{21i}.P_{0i}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nn} & \sum_{i=1}^{n} x_{ni}.P_{0i}\\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1\\ x_2\\ x_3\\ \vdots\\ x_n\\ 1 \end{pmatrix} \qquad (7)$$

where P0 = (P_{01}, P_{02}, …, P_{0n}) is the coordinate of the origin of the new frame.

Similar to previous, one can move from one coordinate to the other by simple manipulation of the above matrix equation. The coordinates of the point in world given its coordinates in viewing is

$$\begin{pmatrix} x_1\\ x_2\\ \vdots\\ x_n\\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & P_{01}\\ 0 & 1 & \cdots & 0 & P_{02}\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ 0 & 0 & \cdots & 1 & P_{0n}\\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} & 0\\ x_{12} & x_{22} & \cdots & x_{n2} & 0\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ x_{1n} & x_{2n} & \cdots & x_{nn} & 0\\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_1\\ c_2\\ \vdots\\ c_n\\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} & P_{01}\\ x_{12} & x_{22} & \cdots & x_{n2} & P_{02}\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ x_{1n} & x_{2n} & \cdots & x_{nn} & P_{0n}\\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_1\\ c_2\\ \vdots\\ c_n\\ 1 \end{pmatrix} \qquad (8)$$

3 The OpenGL "gluLookAt" Function

In this section and the next section we look at detailed application of the above equations in animation in terms of *incremental* changes of the coordinate systems as a viewer moves around a given object in circular or spherical motion. Hence we work with many frames instead of two. Our objective would be, as the view is changed, to create all the needed frames, compute the coordinates of the object in the new frame, and display the object based on the new coordinates.

We simulate animation by fixing the object in space and viewing it through different coordinate systems. For finer control over the view, the viewing frames are built relative to existing frames. First we generate the needed mathematical framework.

Assume we start with a coordinate system aligned with world coordinates, CS_0. Assume as well we generate a new coordinate system, call CS_k, through generation of several coordinate systems in between call CS_1, CS_2, ….., $CS_{(K-1)}$, in that order, as shown in Figure 2.

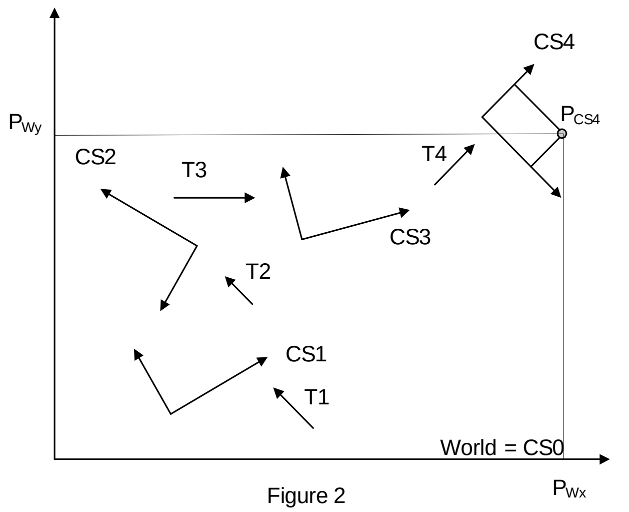


Figure 2

In the figure we assume coordinate system 4 (CS4) is formed by the application of a sequence of intermediate coordinate frames (CS1, CS2, and CS3) where each coordinate system, CS_i, is defined *relative* to the previous coordinate system, $CS_{(i-1)}$. Hence, the origin and basis for CS_1 are defined in terms of the world coordinates. Similarly, the origin and basis for CS3 are defined in the CS2 coordinates frame. Our objective is to compute PW = (P_{wx}, P_{wy}, 1) from knowledge of transformation matrices, Ti's, that move coordinate frame i – 1 to coordinate frame i and from knowledge of the coordinates of a point, P_{CS4}, in frame 4. We would like to compute T in $P_W = T . P_{CS4}$. Using the transformation $P_{CS(i-1)} = T_i . P_{CSi}$ we have $P_{CS3} = T_4 . P_{CS4}$, $P_{CS2} = T_3 . P_{CS3}$, $P_{CS1} = T_2 . P_{CS2}$, $P_W = P_{CS0} = T_1 . P_{CS1}$. Hence

$$P_W = (T_1 . T_2 . T_3 . T_4) . P_{CS4} \text{ or } T = T_1 . T_2 . T_3 . T_4 \qquad (9)$$

To form the final coordinate transformation matrix then we post-multiply the matrices corresponding to frame generation in the same order the coordinate frames are generated. When we consider rendering, we are normally given the point in world coordinate; i.e, we are given point Pw. Based on Pw and a viewing frame, the coordinates of Pw, in the viewing frame, Pv, are computed. This point is then rendered based on other transformations in the graphics pipeline. The point, Pv, can be obtained from equation 9 as

$$P_{CS4} = (T_1 . T_2 . T_3 . T_4)^{-1} . P_w \text{ or } T = (T_1 . T_2 . T_3 . T_4)^{-1} \qquad (10)$$

That is to form Pv (P_{CS4} above), we pre-multiply the transformation for the last frame $(T_4)^{-1}$ defined relative to frame 3 by the cumulative previous transformations for earlier frames.

In OpenGL, a frame can be generated by invoking the graphics utility library function "gluLookAt(P_{0x}, P_{0y}, P_{0z}, P_{refx}, P_{refy}, P_{refz}, V_{upx}, V_{upy}, P_{upz}). Here the point (P_{0x}, P_{0y}, P_{0z}) corresponds to the origin of the viewing frame in world. The orthonormal viewing vectors are computed from the gluLookAt function and are based on the equations

$$V_3 = \frac{V}{|V|}, \quad V_1 = \frac{V_3 \times (V_{upx}, V_{upy}, V_{upz})}{|V_3 \times i|}, \quad V_2 = V_3 \times V_1 \qquad (11)$$

where | | represents the length of a vector and V = (P_{0x}, P_{0y}, P_{0z}) – (P_{refx}, P_{refy}, P_{refz}). In the equation, V_1, V_2, and V_3 correspond to the coordinates of the viewing vectors in world and the set {V1, V2, V3} forms an alternative basis for the set R^3.

Now suppose we have started animation by generating frame, i, using the gluLookAt function. To generate additional frames incrementally we need to define motion relative to frame, i, and accordingly generate frame i + 1. From equation (10), to obtain the coordinates of a point in frame i +1 we *pre-multiply* the new transformation matrix by the cumulative transformation generated from previous frames. Unfortunately, this is not easily done in OpenGL since the transformation generated from cumulative LookAt functions are post-multiplied by the current transformation matrix (CT). We next look at alternative solutions to the above.

4 Incremental Viewing Frame Generation

We consider three solutions: 1) build a library of transformation matrices (here we accumulate (compose) matrices by pre-multiplying the transformation of a new frame by the cumulative transformations), 2) by making use of the function graphics library glLoadMatrix(), and 3) by incorporating the gluLookAt and the glLoadMatrix().

1) Build a Library of transformation matrices: In this method, we construct the needed matrix operations (multiply, translate, etc.) and accumulate results by pre-multiplying a new matrix generation by the cumulative matrix. Once the cumulative matrix is obtained, we have to modify the object description by: a) computing the transformed vertices as (seen in viewing), and b) rendering the points.

2) Making use of the glLoadMatrix: In this solution we keep a copy of the cumulative matrix. On generating a new frame, we: a) compute the viewing transformation for the new frame relative to the previous frame, b) form a new composite matrix by pre-multiplying the generated matrix with the composite matrix, and c) set the current Model-View matrix of OpenGL to the newly generated matrix using the glLoadMatrix. Here, unlike the previous method, we do not need to compute the new viewing coordinates of an object before rendering them. Instead, when we render the object, OpenGL does the transformation by first pre-multiplying each vertex by the current transformation matrix.

3) Incorporating the gluLookAt and the glLoadMatrix(): This option is similar to option 2 but we start by making use of the gluLookAt function as follows: a) in a separate routine we accumulate the transformation matrices similar to method 2 above but with the exception of the initial frame, b) we set the current transformation matrix to the computed accumulated matrix, and c) we use the gluLookAt for the initial frame. OpenGL then post-multiply CT by the matrix generated from gluLookAt. And, as result, produce the correct order of transformations (last to first).

Example: To simplify our analysis we consider an example where an object (a cube) is defined as centered in world coordinates with the initial viewing frame placed at world coordinates (P_{0x}, P_{0y}, P_{0z}) = (0, 0, 1), (P_{refx}, P_{refy}, P_{refz}) = (0, 0, 0) and (V_{upx}, V_{upy}, P_{upz}) = (0, 1, 0); i.e., the viewing coordinates are the same as the world coordinates but with origin translated along the z axis (z = 1). We will simulate moving about the object by generating frames about a unit circle generated by turning about the x-axis in small angles dtheta[2]. Consider Figure 3.

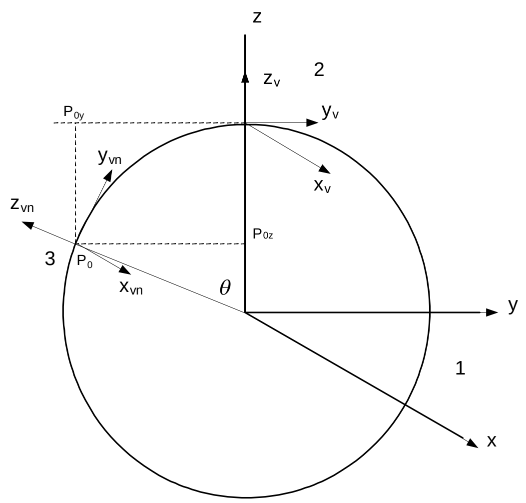


Figure 3

The figure shows three coordinates systems: of the three 2 are viewing coordinates (labeled 2 and 3); the third is the original world coordinate (labeled 1). In the figure viewing coordinates 2 was obtained from viewing coordinates 1 by counterclockwise rotation by angle theta about the x-axis. In the diagram we chose large theta for the purpose of computation the origin and vectors geometrically. Given a point in world coordinates; our task is to generate viewing coordinates by successive small rotations. Hence once a rotation of viewing coordinates is performed, we need to determine the coordinates of the new orthonormal bases and the coordinates of the new origin relative to the previous viewing coordinates. From the figure the coordinates of the new origin are computed as

$$P_{0x} = 0$$
$$P_{oy} = -\sin\theta$$
$$P_{0z} = -1 + \cos\theta$$

The look at point remains the origin which has coordinates (0, 0, –1) in viewing frame 2. The coordinates of the basis for z_{vn} are read from the figure as (0, –sinθ, cosθ). The y_{vn} basis is obtained as a right-turn from z_{vn}, y_{vn} = (0, cosθ, sinθ). With the above we could form the transformation matrix from viewing 2 to viewing 3 as

$$T = \begin{pmatrix} x_{vn} & -x_{vn} \cdot p_{on} \\ y_{vn} & -y_{vn} \cdot p_{on} \\ z_{vn} & -z_{vn} \cdot p_{on} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta & -1+\cos\theta \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

The above matrix is used to compute the coordinates of a point in frame 3 given its coordinates in frame 2.

OpenGL realization: In the OpenGL model-view matrix mode, applying a transformation, T, followed by rendering an object, O, causes a transformation of the vertices of the object based on the equation

$$O_{new} = (CT) \,.\, (T) \,.\, O$$

where CT is the current transformation matrix. As a result, the order of applying transformation in model-view is to first set the CT matrix to the viewing matrix, V, second set the modeling matrix, M, and third call the rendering of a given object. This results in the desired product V . M . O. The order, however, makes it difficult to realize the alternative analogy of rendering with the viewing coordinates changing instead of the set object description. This makes it difficult to realize incremental changes to viewing frames where the coordinates of a new frame are defined in terms of the coordinates of current frame. What is desired is a pre-multiply of new viewing, V1 matrix, by current viewing, CT = V2 matrix, V1 . CT. The gluLookAt function fixes the CT matrix to viewing V2 based on equation 11. Incremental changes using additional gluLookAt functions results in post-multiply of new viewing V2.

In the rendering of example 2, we considered the previous proposed three solutions.

Procedure 1: In this procedure we do not use the gluLookAt function; instead we developed a matrix of geometric operations, functions, (rotate, translate, scale and accumulate for example). The accumulate function causes the user-defined current matrix to be pre-multiplied by a desired transformation such rotation, for example. To render an object based on the accumulated matrix, we first changed the vertices of the object by pre-multiplying each by the accumulated matrix.

Procedure 2: We followed a procedure similar to procedure 2. Here, however, we made use of the OpenGL glLoadMatrixf(). The advantage of this method is in removing the need to write code to change the vertices of the object to be rendered; OpenGL accomplishes this. Using the glutKeyboardFunc(), a new viewing frame V(i+1) defined relative to Vi was computed based on equation 12. Keyboard presses generated new viewing frames. We compute the initial viewing frame V2 (origin of viewing at

[2] It is well known in computer graphics, the effects of generating the frames and rendering are the same as rotating the abject about the x-axis in world. The discussion is intended to provide alternatives that require mathematical analysis and better understanding of OpenGL transformations and graphics pipeline.

(0, 0, 1)) and make the current transformation matrix through the glLoadMatrixf(). To render an object, the object is first transformed based on the given CT matrix. Now to display an object based on viewing V3 defined relative to V2, we: 1) computed the new viewing transformation matrix, V3, using equation 7 (this results in equation 12), 2) formed the matrix product V = V3 . V2, and 3) made V = CT using the glLoadMatrix() function. Subsequent incremental changes in viewing frames can be accomplished similarly.

Procedure3: In this procedure we updated the viewing matrix as done in procedure 2. We first computed the *new* viewing frame matrix and made it CT. We then invoked the gluLookAt() function with parameters that generated the viewing matrix V2. This produces the correct order of matrix multiplication CT = CT . V2 = V3.V2.

A realization of the three procedures on a simple cube with the keyboard used to create new viewing frames based on equation 12, produced the correct result.

5 Conclusion

In this paper we looked at transformations in computer graphics as related to vector spaces from linear algebra in the context of OpenGL. In earlier related work we developed the theoretical framework between transformations and vector spaces. We looked at applications in OpenGL and the concept of incremental change in viewing coordinates. The OpenGL viewing pipeline is built on the concept of current transformation and a post-multiplication of a new transformation by the current transformation matrix (CT). This, however, is not suitable when we look at the alternative interpretation of rendering in computer graphics, where the viewing coordinates change instead of the object world coordinates. We considered several solutions that meet the alternative interpretation.

References:

[1] E. Angel. "Interactive Computer Graphics, A Top-Down Approach Using OpenGL, 5th Edition". Addison Wesley, 2009.

[2] E. Angel. "OpenGL, A Primer, 2nd Edition". Addison Wesley, 2005.

[3] H. Farhat. "Transformations and Basis in the Computer Graphics Viewing Pipeline," International Conference on Frontiers in Education, FECS 2009, pp. 267-271.

[4] D. Hearn, M. Baker, W. Carithers. "Computer Graphics with OpenGL, 4th Edition". Prentice Hall, 2011.

[5] F. Hill, S. Kelly. "Computer Graphics using OpenGL, 3rd Edition". Prentice Hall, 2007.

[6] D. Rogers, J. Adams. "Mathematical Elements for Computer Graphics, 2nd edition". McGraw Hill, 1990.

[7] Foley, van Dam, Feiner, Hughes. "Computer Graphics Principles and Practice, 2nd edition". Addison Wesley, 1990.

TARA Application for Time Analysis of Recursive Algorithms

Irena Pevac, Ramya Gadicharla, Olugbenga Odesina, Sowmya Chaganti
Central Connecticut State University, New Britain, CT

Abstract: *TARA is web based application for learning and practicing time efficiency analysis of recursive algorithms. The application is intended to be used both by instructors and students. Students can practice this traditionally difficult topic through a sequence of sessions. Depending on their background, available time, and expertise, they can work on one or more problems of their choice. Algorithm examples can be chosen based on the algorithm type, difficulty level and problem description. Learners are asked to answer multiple choice questions pertaining to time analysis of selected examples. The system grades each question and records the result in the database. Instructors can review particular student practice scores and the result summary for the entire class to assess their proficiency levels. This enables instructors to assign appropriate practice examples to their students. Instructors can customize TARA by adding new examples to fit learners' background or subject of study. TARA is prerecorded with a set of examples from ESRATEA. The system provides instructors with options to add new algorithm types. It also allows instructors to create additional algorithm analysis questions corresponding to existing and newly created algorithm types. TARA also provides features for instructors to view, modify, and delete the algorithm questions created by them.*

Keywords: Algorithms, Time Performance Analysis, Teaching

1 Introduction

Time efficiency analysis of algorithms is a difficult topic in the algorithms course.

Time analysis of recursive algorithms is more difficult compared to the analysis of iterative algorithms because of the implicit relationship between the time needed to analyze the original problem and the time to analyze one or more sub-problems.

In this paper we discuss the mathematical approach to analyze time efficiency which is independent of the implementation and machine performance details and is used in algorithm course textbooks [1] to [4], [6] to [8] , [10], and [12].

The students in the algorithms course typically experience difficulty performing time analysis which was the motivation for building this software. The application TARA is designed to aid students taking the algorithms course to practice time efficiency analysis of recursive algorithms. TARA is based on ESRATEA [11], and it is currently using the same set of recursive algorithms.

ESRATEA is also designed for students to analyze the time efficiency of recursive algorithms. The application integrates the following three types of recursive algorithms: divide and conquer, chip and conquer, and chip and be conquered. Each of these three types has a multitude of algorithm examples from which the system randomly selects one for students to practice.

The learner starts by selecting one of the three algorithm types and the system selects one algorithm example from the list randomly and displays it. A new window is opened which provides learner with the description of the problem, algorithm code and options to verify his/her answers for time efficiency analysis such as: basic operation, problem size, recurrences, solution for recurrences, and derivation of the solution. Once the student provides appropriate answers, the system evaluates and grades them. In addition, each algorithm is associated with a time performance equivalent graphical example, and its complete analysis, which provides visual illustration of the analysis.

2 TARA versus ESRATEA

ESRATEA is developed as a stand-alone system that uses predefined set of algorithm types and a prerecorded list of algorithm examples for each of those algorithm types.

New algorithm examples and types can only be added by the application designer in ESRATEA. Instructors are not provided with the option to add new examples to the system. For each algorithm feature like basic operation, problem size etc. there is fixed limit for the number of correct and incorrect options that are displayed to learner as possible choices.

In the practice session, the system randomly selects one question at a time. The learner is not allowed to choose questions he/she wants to practice. ESRATEA grades the question during the practice session. The system, however, does not provide any facility to store student's test information for the instructor to review. It is strictly designed for practicing purposes.

TARA is developed as a web-based system. When deployed on a server instructors and students can remotely access the system.

The system allows instructors to introduce additional algorithm types and to add examples in addition to the set currently loaded into the database.

When logged into the system learners are provided with all the available algorithm examples in the database at which point they can practice questions of their choice or work on questions assigned by the instructor.

Instructors can review the practice scores enabling them to identify the problem areas corresponding to each student individually, and to the class as a whole.

3 TARA outline

TARA is designed as a Java web-based system with JSP [5] as front-end. The associated database is designed in MySQL [9]. The database is used to store algorithm examples created by the instructors and information related to test or practice attempts by students. TARA demonstrates two completely different perspectives depending whether user is an instructor or a student.

3.1 TARA for instructors

TARA facilitates instructors' needs to teach and assess students' knowledge of time efficiency analysis. It provides an initial collection of algorithm analysis problems for variety of chip and conquer, chip and be conquered and divide and conquer types that can be used for assessment. TARA also provides features for instructors to create new algorithm types, and to add new algorithm analysis examples, in addition to the ones provided by the application.

Initially instructors can login into the system by using provided user name and password as shown in Figure 1.

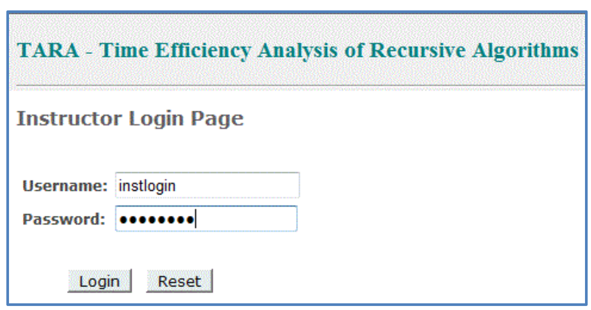


Figure 1

Once logged in, the instructor is allowed to change the password. Updating his/her first name, last name and university in the system is optional.

When the user is logged into the system, the TARA homepage is displayed with various operations the instructor can perform.

The instructor can choose the operation by clicking the corresponding button on the instructor's homepage. See Figure 2.

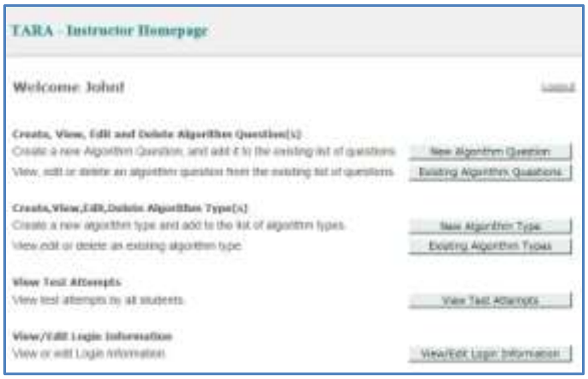


Figure 2

The system is preloaded with a set of algorithm examples which cannot be modified or deleted by the instructor.

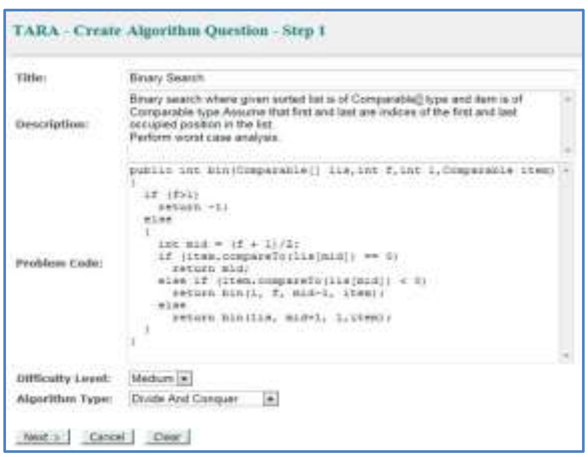


Figure 3

Instructors add an algorithm example by clicking the New Algorithm Question button in the homepage. At that point they can enter the algorithm title, description, upload the problem code, and select difficulty level and algorithm type as shown in Figure 3.

Next, the system takes instructor to the step of creating an algorithm example.

In this page, instructors enter both, a list of incorrect and a list of correct options corresponding to the algorithm properties that the students are going to be tested for as shown in Figure 4.

Figure 4

The instructor also specifies one or more solutions corresponding to the example by selecting appropriate operations from the list displayed as shown in Figure 5.

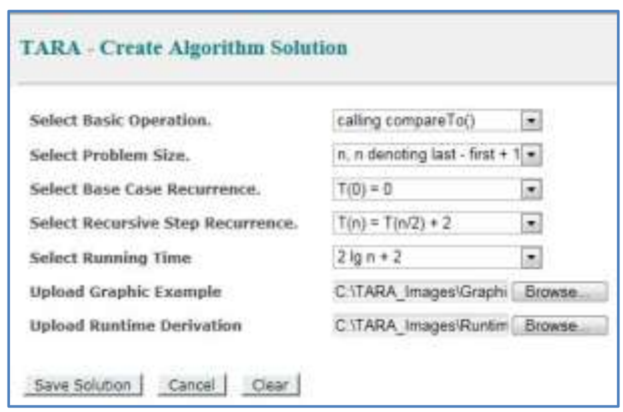


Figure 5

In addition to all options that have to be entered, the instructor should upload corresponding graphical example with time analysis for it as an image, and runtime derivation for the solution of recurrences as shown in Figure 5.

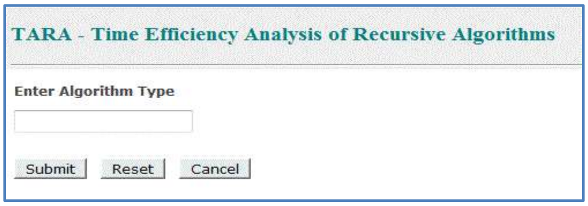
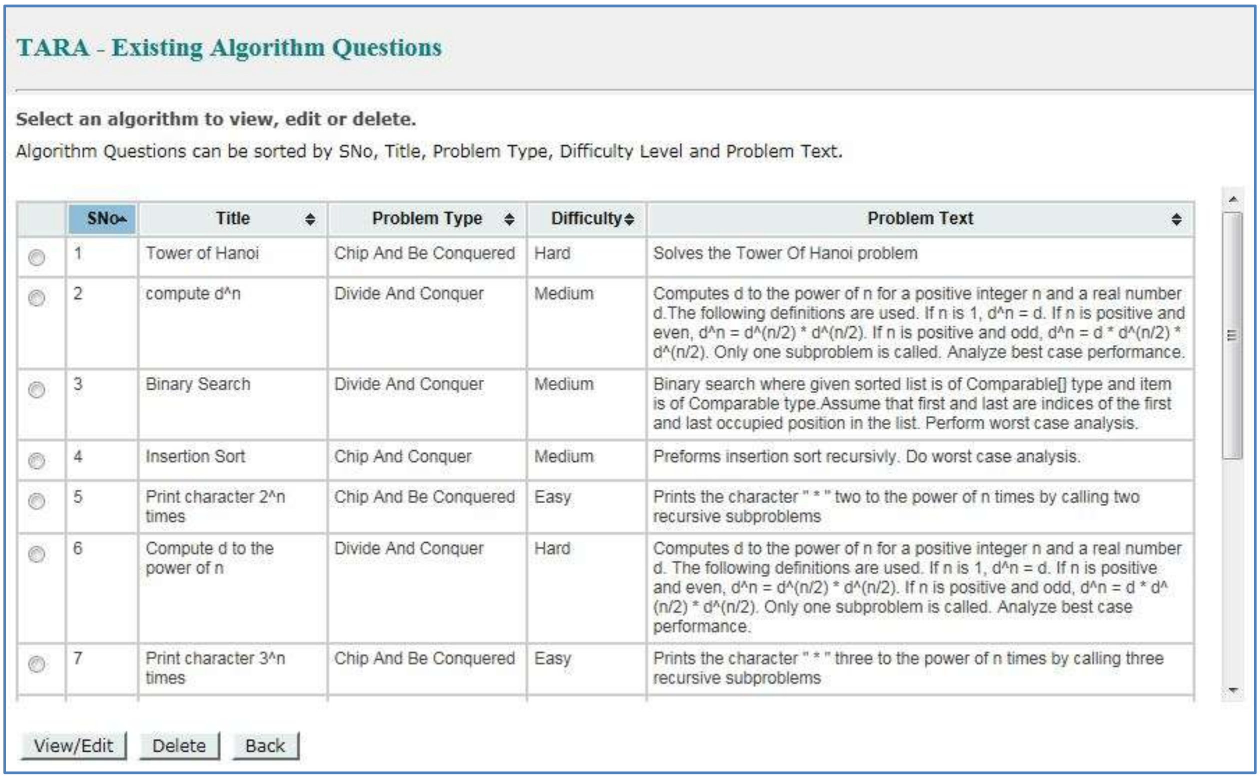


Figure 6

The instructor can view all the available algorithms in the system by clicking Existing Algorithms button in the homepage. This displays the list of both preloaded and user created algorithms as shown in Figure 7.



	SNo	Title	Problem Type	Difficulty	Problem Text
	1	Tower of Hanoi	Chip And Be Conquered	Hard	Solves the Tower Of Hanoi problem
	2	compute d^n	Divide And Conquer	Medium	Computes d to the power of n for a positive integer n and a real number d.The following definitions are used. If n is 1, d^n = d. If n is positive and even, d^n = d^(n/2) * d^(n/2). If n is positive and odd, d^n = d * d^(n/2) * d^(n/2). Only one subproblem is called. Analyze best case performance.
	3	Binary Search	Divide And Conquer	Medium	Binary search where given sorted list is of Comparable[] type and item is of Comparable type.Assume that first and last are indices of the first and last occupied position in the list. Perform worst case analysis.
	4	Insertion Sort	Chip And Conquer	Medium	Preforms insertion sort recursivly. Do worst case analysis.
	5	Print character 2^n times	Chip And Be Conquered	Easy	Prints the character " * " two to the power of n times by calling two recursive subproblems
	6	Compute d to the power of n	Divide And Conquer	Hard	Computes d to the power of n for a positive integer n and a real number d. The following definitions are used. If n is 1, d^n = d. If n is positive and even, d^n = d^(n/2) * d^(n/2). If n is positive and odd, d^n = d * d^(n/2) * d^(n/2). Only one subproblem is called. Analyze best case performance.
	7	Print character 3^n times	Chip And Be Conquered	Easy	Prints the character " * " three to the power of n times by calling three recursive subproblems

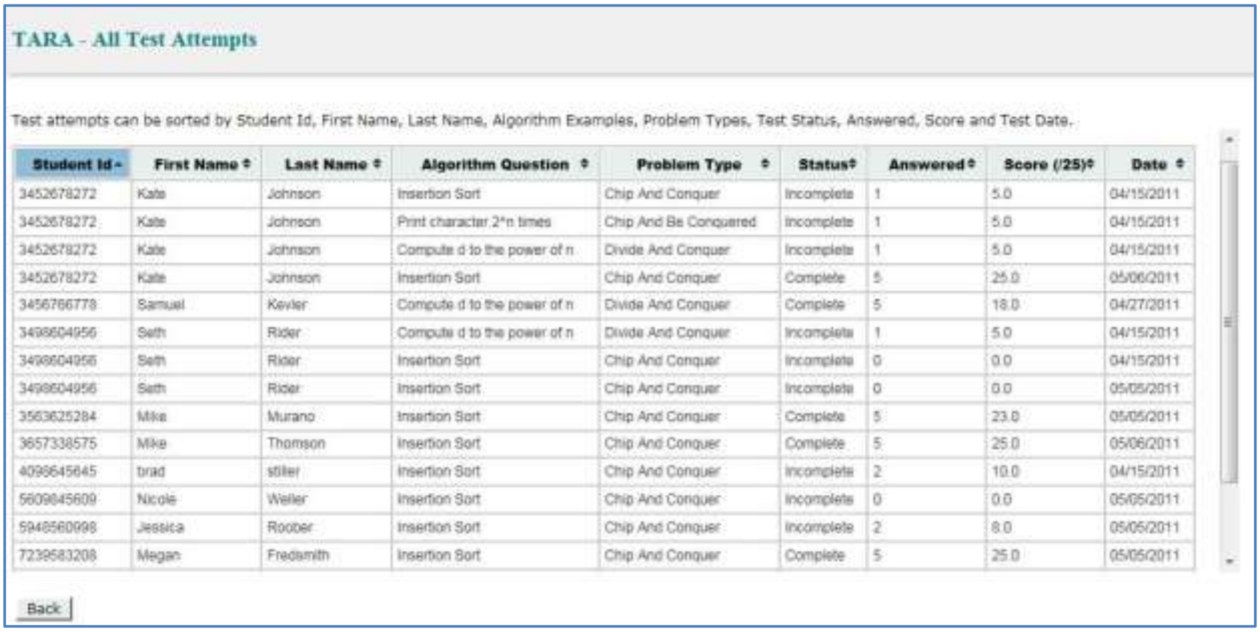
Figure 7

The instructor can edit or delete an algorithm by choosing it from the list of available algorithms for edit or delete. The user is not allowed to delete or modify default set of algorithms.

The system provides three algorithm types and the instructors can create more algorithm types in addition to these. To create an algorithm type, the instructor can click on New Algorithm Type button in the home page and enter the algorithm type as shown in Figure 2.

To modify the algorithm type, the instructor can click on the Existing Algorithm Types button which displays the list of available algorithm types in the database and the user can choose the algorithm type to modify.

To view all test attempts performed by all students, the instructor can click on the View All Test Attempts button which displays a list of all student ids, names, scores, test dates and number of questions completed as shown in Figure 8. The instructor can sort this information based on ids, scores, names etc. This allows the instructors to analyze the progress of the class. Analogous results can be obtained for an individual student's practicing session.



TARA - All Test Attempts

Test attempts can be sorted by Student Id, First Name, Last Name, Algorithm Examples, Problem Types, Test Status, Answered, Score and Test Date.

Student Id	First Name	Last Name	Algorithm Question	Problem Type	Status	Answered	Score (/25)	Date
3452678272	Kate	Johnson	Insertion Sort	Chip And Conquer	Incomplete	1	5.0	04/15/2011
3452678272	Kate	Johnson	Print character 2^n times	Chip And Be Conquered	Incomplete	1	5.0	04/15/2011
3452678272	Kate	Johnson	Compute d to the power of n	Divide And Conquer	Incomplete	1	5.0	04/15/2011
3452678272	Kate	Johnson	Insertion Sort	Chip And Conquer	Complete	5	25.0	05/06/2011
3456766778	Samuel	Kevler	Compute d to the power of n	Divide And Conquer	Complete	5	18.0	04/27/2011
3498604956	Seth	Rider	Compute d to the power of n	Divide And Conquer	Incomplete	1	5.0	04/15/2011
3498604956	Seth	Rider	Insertion Sort	Chip And Conquer	Incomplete	0	0.0	04/15/2011
3498604956	Seth	Rider	Insertion Sort	Chip And Conquer	Incomplete	0	0.0	05/05/2011
3563625284	Mike	Murano	Insertion Sort	Chip And Conquer	Complete	5	23.0	05/05/2011
3657338575	Mike	Thomson	Insertion Sort	Chip And Conquer	Complete	5	25.0	05/06/2011
4098645645	brad	stiler	Insertion Sort	Chip And Conquer	Incomplete	2	10.0	04/15/2011
5609845609	Nicole	Weller	Insertion Sort	Chip And Conquer	Incomplete	0	0.0	05/05/2011
5948560998	Jessica	Roober	Insertion Sort	Chip And Conquer	Incomplete	2	8.0	05/05/2011
7239583208	Megan	Fredsmith	Insertion Sort	Chip And Conquer	Complete	5	25.0	05/05/2011

Back

Figure 8

3.2 TARA for Students

Students login into the system by entering their student id, first name and last name. When students login into the system, TARA displays a list of available algorithm examples in the system. Fraction of that list is shown in Figure 9. Students can sort the list by algorithm type, title, difficulty level, and by other algorithm characteristics.

Figure 9

Students choose the type of the algorithm they want to practice and click solve to practice the algorithm example. This opens a practice/test window as shown in Figure 10. It displays the question title, algorithm code and multiple choice answer options. Grading scheme implemented in TARA is similar to that of ESRATEA. The students proceed by selecting appropriate answers to algorithm properties in the order of basic operation, problem size, base case recurrence, recursive step recurrence, and solution for recurrence.

Students are only allowed to select the next option if the previously selected option is correct. The score is calculated for each selection and for each attempt, and it is displayed to the student as he proceeds with the practice or test.

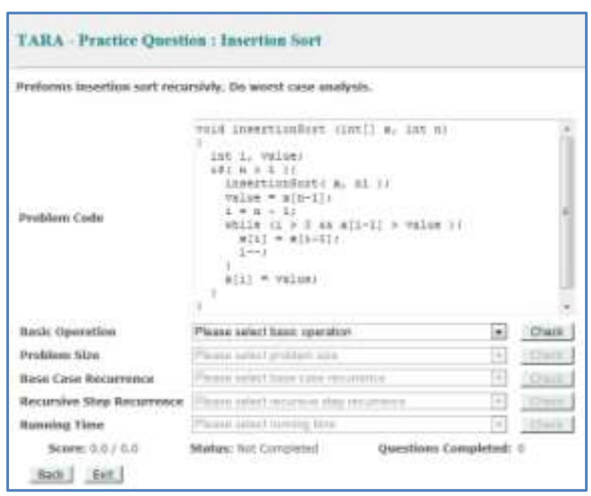


Figure 10

Once the test is completed as shown in Figure 11, the student is given an option to view the graphic example and runtime derivation associated with the analysis.

Each test attempt along with the student information is recorded into the database and can be reviewed by the instructor.

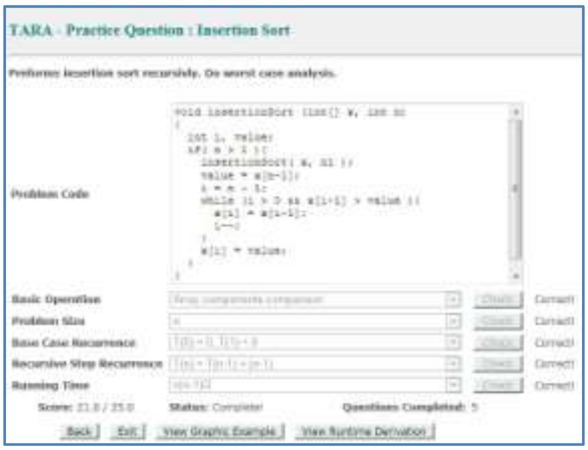


Figure 11

4 Future Work

The system is intended to be distributed to the instructors teaching algorithms courses and willing to use it as a learning tool for students in their courses. Upon more extensive use by students, research can be conducted to determine the usefulness of the system.

Further improvements to the system like generating various types of reports of student performance, assesment of scores of various student groups on selected algorithm questions can be done.

Performance assessment is planned before and after using TARA. This is done by comparing scores on recursive algorithm time efficiency analysis among students in the class that used TARA against those students that did not use TARA.

References

[1] Aho, A. V., Ullman, J. D. (1995). *Foundations of Computer Science* (2nd ed.). Computer Science Press.

[2] Baase, S., Van Gelder, A. (2000). *Computer Algorithms* (3rd ed.). Addison Wesley Longman.

[3] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Clifford, S. (2007). *Introduction to Algorithms* (2nd ed.). MIT Press; McGraw-Hill Book Company.

[4] Goodrich M., Tamassia R. (2002). *Algorithm Design: Foundations, Analysis, and Internet Examples* John Wiley & Sons.

[5] Java Servlet Technology – Oracle, http://www.oracle.com/technetwork/java/index-jsp-135475.html.

[6] Johnsonbaugh R. Schaefer M. *Algorithms* (2004). Pearson Prentice Hall.

[7] Levitin, A. (2007). *The Design & Analysis of Algorithms* (2nd ed.). Pearson - Addison Wesley.

[8] McConnell, J. J. (2008). *Analysis of Algorithms* (2nd ed.). Jones and Bartlett Publishers.

[9] MySQL, http://www.mysql.com/.

[10] Neapolitan R. Naimipour K. (2004). *Foundations of Algorithms Using Java Pseudocode* Jones and Bartlett.

[11] Pevac, I. ESRATEA Educational Software for Recursive Algorithm Time Efficiency Analysis *WORLDCOMP'10 Proc. of the 2010 Int. Conf. on Frontiers in Education: Computer Science & Computer Engineering*, (Ed. Arabnia, Clincy), Las Vegas, NV, SCREA Press, 2010, pages 367-373.

[12] Sedgewick R. (2011) *Algorithms* (4th ed.). Addison Wesley.

Rhetoric for Synchronous video communication

Chrysoula Themeli, Educational Research, Hellenic –American University, Athens, Greece

Justin Bonzo, Educational Research, Lancaster University, Lancaster, United kingdom

Abstract - *Synchronous video communication (SVC) offers a promising way of providing opportunity for synchronous conversation and learning between members of a group that is separated by distance. The perceived benefits of SVC are: presence, immediate feedback, access, cost, and interaction. This presentation will look at the perceived benefits through the research lens of participatory and appreciative action and reflection. The concepts of rhetoric, ethos, pathos and logos will be used to illustrate the possibility of research praxis. It is hoped that this will help generate knowledge of this medium and the possible rhetorical factors that may influence the pedagogy behind SVC.*

Keywords: rhetorics, logos, ethos, pathos.

1 Introduction

Synchronous video communication (SVC) - synonymous with video conference or Interactive Videoconference - is defined as real-time, video-enhanced conversation, an immediate give and take, between at least two participants in different locations (Alexander, Higgison, & Mogey, 1999; Chandler & Hanrahan, 2000; Gibson & Cohen, 2003; Suthers, 2001). The prominent characteristic feature of SVC is the immediate exchange of information and sharing of facilities among distant users (Brown, 2001; Finn, Sellen, & Wilbur 1997; Andrews, T. & Klease, G. 1998; Gürer, et all 1999). It is opposite of webcasting, which is dominated by lectures *in situ* (Haga & Kaneda, 2005). The introduction of the videoconference facility to the education system some 20-years-ago was quickly recognized as having the potential to resolve difficulties facing distance education (Gillies, 2008, p.108) Since then, more and more people are using videoconferencing to communicate, teach, learn, and facilitate their education or research in many fields (Gillies, 2008; Chia et al, 2009 ;Sedgwick & Spiers , 2009).

Though distance education is expanding in all continents, SVC has not been an option thoroughly investigated (Saw et al, 2008; Knipe & Lee, 2002).

> "However, due to the current commitment to educational convenience that has been institutionalized in asynchronous distance education programs, little, if any, progress in integrating audio and videoconferencing tools into distance education has been made" (Corbeil, 2006) p.388).

There is a great deal of optimism about the potential of distance education technology (Larreamendy-Joerns & Leinhardt, 2006) and, until recently, this type of instruction has typically been asynchronous in nature. That is to say, a greater integration of videoconferencing with open learning environments and other social media tools could help make better sense to learners (Tomadaki et al, 2008). Hrastinski (2008) maintains that users decide how, why and when to use a medium and their preferences need to be taking seriously while designing online courses. Research findings in relation to outcomes and meeting educational needs in using SVC are mostly consistent, while findings about satisfaction among educators and students are inconclusive (Berger et al. 2009, p.478). Therefore, SVC deserves pedagogical reinvention and further inquiry (Anastasiades, 2006). The aims of this study are:

- to acknowledge what users perceive as the strength of the SVC
- to identify rhetorical appeals as factors that could enhance a holistic pedagogy
- To reinvent a new holistic pedagogy so as to improve future praxis of SVC

2 Literature review

Benefits of SVC: Its perceived benefits can be summarized in terms of, presence, immediate feedback, access, cost, and interaction.

1. Presence - Living the moment of learning

Real time video communication could promote teaching, and cognitive and social presence. Garrison, Anderson, and Archer (2000) argue that three aspects of computer-mediated communication are needed for any educational transaction: cognitive presence, social presence and teaching presence. To illustrate, cognitive presence can be defined as the learning process though which learners negotiate meaning and construct new knowledge. Short and colleagues (1976) describe social presence as the ability of a medium to allow people to experience the actual presence of a communicator. Gillies (2008) found out that social presence is the element that should be maximized during videoconference sessions. Gunawardena and Zittle (1997) related *intimacy* and *immediacy* to social presence, claiming that feelings associated with social presence is a predictor of learner satisfaction in online environments. Last but equally important, teaching presence can be seen as tutors' work of instruction inspiration and motivation in order to guide and support learners 'experiences.

As real time video links enhance the presence experience (Lombard and Ditton, 1997), videoconferencing has been deployed in physically distributed workgroups around the world. While face-to-face communication is rich because it includes deictic elements and objects visible to both participants of the communication (Nardi, 2005), videoconferencing entails the most essential elements of a meeting (Townsend et al, 2002). In addition, a number of research studies concluded that nonverbal and verbal cues fostered all kinds of benefits : more learning, more student motivation, and more empowerment (Freitas, 1998, pp. 366-67; Haefner, 2000; Chapman, Uggerslev, et al., 2003). Another important role of media is to provide valuable 'cues' about the presence of others, including facial expressions and other key aspects of presence (Short et al, 1976). Gillies (2008) refers to the studies of many authors (Hills, 2005; Ryan, Scott, Freeman, & Patel, 2000;Smyth, 2005) who claim that the improvements brought by bandwidth links have lifted many barriers. It means that the quality of video is now such that it removes many of the issues of facial expression and body language which previously limited the degree of 'social presence' possible in the videoconference.

2. Immediate feedback

Secondly, it provides immediate feedback. Videoconferencing has made it possible for students to feel as if they are part of a real classroom-learning environment by providing immediate contact, motivation, and clarification of meaning (Steeples, Jones, & Goodyear, 2002). De Freitas and Neumann (2009) refers to the paper of Schullo *et al*, (2005), which is focused upon distance education, and argues that there are two main challenges: "ensuring the maximum interaction with groups and a 'lack of confirmed pedagogic strategy for supporting work in synchronous environments" (Schullo *et al*,2005, p. 3). Their study found that learners who were left behind, were not helped enough by the use of asynchronous methods as "lack of immediacy still makes it difficult for students to connect quickly with each other or their instructor" (Schullo *et al*, p. 3). In addition, they claim that passive modes of delivering content and a lack of active student participation or effective interaction cause more difficulties in distance education cohorts such as high dropout rates, because of limited or no face-to-face contact between students and tutors (de Freitas & Roberts, 2004).

3. Access

Remote access to expert input or opinion is also cited as a benefit of videoconference (Laurillard, 2002, p. 156). This is not only academically advantageous for learners, but is also an economically efficient way for institutions to provide quality staff–student contact (Gilles, 2008, p.108; Ba, & Keisch, 2004). The audience for courses can be increased by teaching face to face with one group and providing access to a second centre elsewhere (Carville & Mitchell 2000).

4. Cost

Videoconferencing has not only allowed U.S. state universities to address their 'mandate for equal access to citizens wherever they were located in the state,' but it also provided them with a cost effective way of doing so (Bates, 2005, p. 180).

5. Interaction

Gillies (2008) refers to literature (Bates, 2005; Fardanesh, 2002; Mason, 1998,Offir & Lev, 2000; Ryan et al., 2000; Smyth, 2005) that points to effective interaction as a prerequisite for successful videoconferencing in education, although only sometimes is the required methodology for this aspect of constructivist thinking fully provided. Studies have shown that videoconferencing, under careful organization, can enhance computer-supported group-based learning. This is an important part of contemporary education, focusing on 'cooperative' and 'collaborative' learning, inspired by collaborative environments similar to original working processes (Strijbos et al, 2003). Compared to other methods of distance education, videoconferencing has promised benefits in terms of real-time interaction, immediacy, motivation, and collaborative learning (Bates, 2005; Brown & Liedholm, 2002; Guri-Rosenblit, 1999; Rosen, 1996; Gilles, 2008). Learners together at a remote site can experience particular social benefits to aid their learning. There is the sense of togetherness and shared experience, a fellowship that can help offset the particular danger of attrition where students study both remotely and individually (Bates, 2005; Wheeler, 2005; Wheeler & Amiotte, 2004). Further social advantages from interaction mostly marked at primary or secondary education levels where it has been used to bring together children and young people from very different linguistic, social, and cultural backgrounds. There is a considerable body of research evidence, which points to the benefits of mutual understanding, broadened awareness, tolerance, and new insights, afforded by videoconference interaction of this kind (Abbott, Austin, Mulkeen, & Mecalfe, 2004; Austin, Smyth, Mallon, Mulkeen, & Metcalfe, 2004; Cifuentes & Murphy, 2002; Comber, Lawson, Gage, Cullum-Henshaw, & Allen, 2004; Jones & Sorenson, 2001; Lewental & Kress, 2005; Martin,2005; Payne, Gooday, Coutts, Duncan, & Wolfe, 2006 Anastasiades, 2009, Gilles,2008). Last but equally important, in the case of web-based courses, enhanced opportunities for active participation of students who are hesitant to express themselves in a traditional classroom setting (Anastasiades et al. 2009)

Challenges and Implications

Perceived implications of the videoconference in educational settings can be summarized as relating primarily to issues of technological obstacles, flexibility, and pedagogy.

1. Technological obstacles

The foremost disadvantages of video conferencing are the technical difficulties associated with smooth transmissions that could result from software, hardware or network failure. Remote connections are sometimes hampered by environmental changes. On some occasions, the absence of technical support personnel creates difficulty for participants who are unfamiliar with the videoconferencing technological concepts. Where there are many individuals involved, the camera may not identify the speaker readily in an interactive setting and so others may need to rely on voice alone, which is limiting. "Issues around sound (the receiving site had to come up to a mic to talk), time delay, and picture quality impeded the instructor's spontaneity and made the lecture "rather stilted" (Carville & Mitchell 2001, p.45).

2. Pedagogy

One of the reputed pedagogical problems of the videoconference is the lecture format. Gilles (2008) draw early evidence in the literature and maintains that videoconferencing was dominated by lecturing (Dallat, Frazer, Livingston, & Robinson, 1992; Freeman, 1998; Mason, 1998; Oliver & Reeves, 1996). The role of the educator using the medium needs to be seen under different light.

Simply transferring 'live' classroom approaches to the videoconference suite is seen as inadequate (Martin, 2005; Ryan et al., 2000; Smyth, 2005; Gilles, 2008).To illustrate, SVC may not maintain the attention of all participants (Reed & Wooduff, 1995).

3. Real-time interaction or no time,

It does obviously restrict the autonomy of the learner because, time zone may cause difficulties and if some participants miss the online session it may not be possible to attend another one. The technological logistics, currently at least, require that distant students have to be present at a site, at a set time, to access locally the programmes coming from a distant provider (Bates, 2005, p. 180)

Curant et al. (2008, p.6) maintain that 'one-size-fits-all' approach to retaining students and providing them with engaging online learning environments is not efficient. They firmly believe that it is not productive to use technology in teaching methods and expect students to use it appropriately (Currant & Whitfield, 2007). Thus, real-time video enhanced learning could be investigated further with the experienced with specific tools and media affordances but students and tutors preferences and digital literacy may play a key-role in the learning process.

Theoretical framework

It is the theory which decides what we can observe.
~ Albert Einstein (1879-1955)

Rhetoric is the study of writing and speaking as a means of communication (Merriam-Webster dictionary), one of the most important of Aristotle's philosophies. Aristotle believed that rhetorical appeals, logos, pathos and ethos help audience understand ideas presented, and move people to new ways of thinking and acting. Ethos is a demonstration of good will towards the audience, good sense of knowledge of the subject at hand and good character. It is also an effort to establish common ground among participants. Logos is the effective use of reason and pathos entails stirring of emotions (Glenn & Gray, 2008 pp. 122-123). Rhetoric for the purposes of this inquiry can be defined as the pedagogy of speaking, writing, and participating in online activities via video communication. The inspiration of Aristotle's rhetorical appeals played a critical role in framing the research questions, without assuming exact parallel with his philosophy.

For this study, rhetoric is a new pedagogy under investigation. Rhetorical appeals (logos pathos and ethos) are considered as contributing factors. Logos in the context of SVC demonstrates the use of reason: asking questions, providing concepts, reframing concepts and generating new ideas. It is closely correlated with cognitive presence. Cognitive presence is defined as the extent to which learners are able to *construct and confirm* meaning through sustained discourse and critical thinking (Garrison et al. 2000, 2001). Both these definitions focus on the construction of meaning through a process of inquiry. This means a shift from "lecturing and telling" to questioning, reflexivity, and open inquiry. After all, Plato's dialogues have been an integral part of teaching and learning. Dialogue and conversation are by definition immediate interchanges that are synchronous (Haefner, 2000). Ethos, in the SVC environment, establishes social harmony and entails participants' good will towards the audience, credibility of information exchanged and empowering participation in a democratic environment (Fung, 2006). It presupposes freedom to negotiate educational objectives and organization. Finally, Pathos demonstrates itself through empathy, authentic understanding, and appreciation for best intentions and praxis. Pathos is partially correlated with the concept of Appreciative intelligence. It is the appreciation the users share with one another while learning.

Trochim (2006) states that there are two realms involved in research—theory and observation. I am interested in finding what are the strengths of SVC. My interest comes about because I have observed that the factors influencing the best practice of the medium can be related to rhetoric, the pedagogy of writing, speaking as a means of communication. Thus, it remains to be tested if this hypothesis is true, to what extent and in what ways. The inspiration of Aristotelian Rhetoric in the foreground of SVC will create an axiology – a set of criteria – to evaluate critically the theoretical assumption, that logos, pathos and ethos as they are defined for the purposes of this project may correlated positively or negatively with the best educational praxis of SVC in the field of distant education.

Overarching Research Questions

1)What are the strengths and best practice of synchronous video communication?
2)To what extent and in what ways rhetoric (living the moment of learning with logos, pathos, and ethos) could affect best praxis SVC?

Rhetoric, Methodology, and praxis for Synchronous video communication (SVC)

Rhetoric as pedagogy defined above, could be implemented in a democratic, participatory and dialogical framework. The methodology aligned is called, participatory and appreciative action and reflection (PAAR) .The label 'participatory and appreciative action and reflection' (PAAR) was first used by Ghaye (2008). It adds a new dimension to participatory action research, Appreciative Intelligence (AI); the ability to acknowledge best practices (Thatchenkery & Metzker, 2006).

Action research projects start from reflecting on a problem, framing, and finally resolving it. Ghaye et al, refer to Loughran (2006), who believes that if the researchers focus only on problems to solve they could easily be influenced by the negative connotations of the word 'problem' which means error of judgment or failure. Ghaye et al,(2008, p.362) draws on the work of Kemmis (2005), who argues that 'changing practice' is a "task of changing such things as discourse in which practices are constructed and the social relationships which constitute practice".

Thus, PAAR is based on identifying the successes and strengths. What makes people and practice better? They (Ghaye et al, 2008) define PAAR as a style of research, which requires researchers to use their "appreciative intelligence, to focus on the best of what is currently experienced, seek out the root causes of this, then design and implement actions that amplify and sustain this success". In the same vein, Zeichner (2001, p.278) acknowledges that in educational action research the focus must change from the student problems to the student resources and accomplishments (p. 278).

The main question PAAR asks is: "what are our successes and how can we amplify them to build and sustain a better future from valued aspect of the positive present?" (p.364). It is described in the framework of appreciative intelligence and multiple intelligences theory Gardner (1993). "Appreciative Intelligence is the ability to perceive the positive inherent generative potential in a given situation and to act purposively to transform the potential to outcomes" (Thatchenkery & Metzker, 2006). Thatchenkery and Metzker (2006) suggested that appreciative intelligence is composed of three characteristics: the ability to appreciate the positive, reframe it and see how the future evolves. It is not only a theoretical approach, it also involves social actions –"the necessary actions to positively engage with others so that valued outcomes unfolds from the generative aspects of the current situation"(Ghaye et al,2008,p. 366).

The participatory and appreciative characteristic (Jacobs, 2006; Ghaye et al, 2008) require all involved to be active, to be explicit about the perspective from which knowledge is created and to see democratic peer relationships as a form of inquiry that serves the practical ethos. PAAR actively draws upon the notion of empowered participation (Fung, 2006) and its associated process of deliberative democracy (Thomson & Gutmann, 2004). "Together these demonstrate a commitment to positively engage with, and provide equal opportunities for, all those involved to participate directly in decisions that affect their own and others welfare" (Ghaye et al,2008p. 368).

The appreciative inquiry of PAAR helps the quest for treasure on the open sea because of the "appreciative intelligence". To explain, it makes the researcher ask questions people feel more comfortable to answer and contribute to. Thatchenkery, in his book 'Appreciative Inquiry and knowledge Management (2007), demonstrates that appreciation – or affirmation – is the key ingredient for people to trust each other and overcome their inhibitions and concerns about sharing what they know. It does not mean though that the project is a utopian adventure. On the contrary, critical reflection and transformative actions are integral part of the practical wisdom of the methodology (Kemmis,2006).

Some standards of PAAR 's judgment (Ghaye et al, 2008, p.375):

1. Inclusivity: How far have all interested participants, collectively, developed their appreciative 'gaze'?
2. Emotional engagement: How far is there evidence of participants being alive to PAAR as emotional work?
3. Understandability: How far are participants demonstrating a commitment to appreciative communicative action?
4. Mutualism: How far is there evidence of interdependence of creative and critical thinking?
5. Transformation: In the reframing of emotions, understanding, and practices, how far do 'new angles of vision' emerge?
6. Communicative freedom: When building practical wisdom, how far is the process ethically and socially justifiable and sustainable?
7. Moral courage: When moving forward, How far is there evidence of committed action to build a 'better' future from significant aspects of positive present?

The seven standards of PAAR can be seen as the logos, ethos and pathos democratic engagement and participation. Aristotelian Rhetoric is deemed to be not only the characteristics of public speakers and political figures but also of every democratic citizen that is aware with what is happening in the world. "A participatory worldview is a political statement, as well as a theory of knowledge that implies democratic, peer relationships as the form of inquiry" (Reason and Bradbury, 2001, p.9). Therefore, theoretical framework, methodology, and Praxis are interrelated in the following table.

THEORETICAL FRAMEWORK	PAAR METHODOLOGY STANDARDS	PRAXIS
Logos	Understandabiliy	Asking questions,

		providing or receiving feedback
	Transformation	Reframing concepts
	Mutuality	Generate new ideas through interaction
Ethos	Moral courage	Credibility of information exchange, responsible participation
	Communicative freedom(social trust)	Good will towards the audience, freedom to negotiation of objectives or organization
Pathos	Inclusivity	Empathy, authentic understanding, belonging to online community
	Emotional engagement	Appreciate different perspectives

Table 1

References

Andrews, T. & Klease, G. (1998). Challenges of multisite videoconferencing: The development of an alternative teaching/learning model. *Australian Journal of Educational Technology,* 14 (2), 88.

Anastasiades, P. (2009). *Interactive Videoconferencing and Collaborative Distance Learning for K-12 Students and Teachers: Theory and Practice* NY: Nova Science Publishers, Inc

Anastasiades, P. (2006). Interactive Videoconferencing in K- 9 Education: "ODUSSEAS 2000-2004" a case study in Elementary Schools in Greece and Cyprus. . *Proceedings of the Diverse 2006, 6th International Conference on Video and Videoconferencing in Education.* 5-7 July, 2006, Scotland , Glasgow: Caledonian University.

Austin, R., Smyth, J., Mallon, M., Mulkeen, A., & Metcalfe, N. (2004). *Dissolving boundaries – Supportingtransformation in the classroom?* Retrieved March 3, 2007, from http://www.dissolvingboundaries.org/research/sep2004.doc

Ba, H., & Keisch, D. (2004). Bridging the Gap Between Formal and Informal Learning: Evaluating the Seatrek Distance Learning Project. Retrieved Febuary 11, 2008, from http://cct.edc.org/report_summary.asp?numPublicationId=177

Bates, A. (2005). *Technology, e-learning and distance education* (2nd ed.). Abingdon, UK: Routledge.

Berger, R.,Stein,L.,& Mullin,J.B (2009).Videoconferencing: A Viable Teaching Strategy for Social Work Education? Social Work Education: The International Journal, 28(5), 476:ddoi:10.1080/02615708023086 25

Brown, S. (2001). Views on Videoconferencing Higher Education and Research Opportunities in the UK (HERO), March Issues - From http://www.hero.ac.uk/inside_he/archive/views_on_videoconferencin883.cfm?

Brown, B., & Liedholm, C. (2002). Can web courses replace the classroom in principles of microeconomics? *American Economic Review, 92*(2), 444–448.

Chandler, G., & Hanrahan, P.. (2000). Teaching using interactive video: creating connections. *Journal of Nursing Education* 39 , pp. 73–80.

Gibson, C., & Cohen, S. (eds.) (2003). *Virtual Teams That Work: Creating Conditions for Effective Virtual Teams.* San Francisco. CA: Jossey-Bass/Wiley.

Chia, R.C.,Poe, E. & Wuensch, K.L. (2009) Attitude Change after Taking a Virtual Global Understanding Course, *International Journal of Social Science,4:2*

Corbeil, J.R. (2006). The (r)evolution of synchronous communication in distance education. *Issues in Information Systems*, *7*(1),

Creswell, J. W., S. Trout and J. E. Barbuto (2004). A decade of mixed methods writings: A retrospective. Retrieved October 7,2009 from http://www.aom.pace.edu/rmd/

Currant, N. & Whitfield, R. (2007) *Students aren't prepared for web 2.0 learning, are they?* [presentation] Association for Learning Technology Conference

Currant, N; Currant, R; Whitfield, R; Hartley, P (2008) DEFINING GENERATION Y: TOWARDS A NEW DIGITAL TYPOLOGY OF LEARNERS in (eds) Lawton et al

Proceedings of the 3 rd European First Year Experience Network Conference , Wolverhampton

Guri-Rosenblit, S. (1999). *Distance and campus universities: Tensions and contradictions.* Kidlington: Elsevier Science.

Glenn, S., & Gray, L. (2008). *The writer's harbrace handbook, brief,* . N.York: Thomson Wadsworth, .

Kemmis, Initials. (2006). 'participatory action research and the public sphere'. *Educational Action Research*, *14*(4), Retrieved from http://dx.doi.org/10.1080/09650790600975593 doi: 10.1080/09650790600975593

Kemmis, S. (2005). Knowing practice: Searching for saliences. *Pedagogy, Culture & Society, 13*(3),391–426.

Freitas, F., Myers, S., & Avtgis, T. (1998). Student Perceptions of Instructor Immediacy in Conventional and Distributed Learning Classrooms. Communication Education 47(4), 366-72.

de Freitas, S. & Roberts, G. (2004). Does distance e-learning work? A comparison between distance and face-to-face learners using e-learning materials. *Association of Learning Technology Journal*, *11*, 2, 69–87..

Finn, K. E., Sellen, A. J., & Wilbur, S. B. (Eds.). (1997). *Video-mediated communication*. Mahwah, NJ: Lawrence Erlbaum Associates.

Carville, S., & Mitchell, D. R. (2001). It's a Bit Like Star Trek: The effectiveness of video conferencing. *Innovations in Education and Training International*, 37(1), 42-49.
.

Garrison, D. R., Anderson, T., & Archer, W. (2000). Critical inquiry in a text-based environment: Computer conferencing in higher education. The Internet and Higher Education, 2(2-3), 87-105.

Ghaye T., Melander-Wikman,A., Kisare, M., Chambers,P., Ulrika, B., Kostenius, C., Lillyman, S.(2008) Participatory and appreciative action and reflection (PAAR) democratizing reflectivepractices. *Reflective Practice* 9(4), 361- 398

Gillies, D.(2008) Student perspectives on videoconferencing in teacher education at a Distance, *Distance Education* 29:1, 107- 118 DOI: 10.1080/01587910802004878

Greene, J. C., Caracelli, V. J., & Graham, W. F. (1989). Toward a conceptual framework for mixed-method evaluation designs. *Educational Evaluation and Policy Analysis, 11,* 255-274

Gürer, D., Kozma, R., & Millán, E. (1999). Impact of shared applications and implications for the design of collaborative learning environments. In S. Lajoie (Ed.), *Proceedings for the ninth world conference on artificial intelligence in education* (AI-ED99, pp. 439-445). Amsterdam: IOS Press.

Gunawardena, C. N., & Zittle, F. J. (1997). Social presence as a predictor of satisfaction within acomputer-mediated conferencing environment. *The American Journal of Distance Education, 11*(3), 8-26.

Haga, H. & Kaneda, S. (2005). A usability survey of a contents-based video retrieval system by combining digital video and an electronic bulletin board. *Internet and Higher Education*, 8, pp.251-262

Haefner, J. (2000). Opinion: The Importance of Being Synchronous. *Academic.Writing*. Retrieved January 10, 2006, from the WorldWide Web:http://wac.colostate.edu/aw/teaching/haefner2000.htm

Hrastinski, S., (2008), Asynchronous and Synchronous E-Learning *EDUCAUSE Quarterly,* vol. 31, no. 4(October–December).

Knipe, D., & Lee, M. (2002). The quality of teaching and learning via videoconferencing. *British Journal of Educational Technology, 33*(3), 301–311.

Kemmis, Initials. (2006). 'participatory action research and the public sphere'. *Educational Action Research*, *14*(4), Retrieved from http://dx.doi.org/10.1080/09650790600975593 doi: 10.1080/09650790600975593

Lewental, M., & Kress, M. (2005). *Making global connections: The virtual classroom project.* Paperpresented at the 38th Annual Association of Small Computer Users in Education (ASCUE)Conference, Myrtle Beach, SC. Retrieved March 15, 2007, from ERIC Document ReproductionService No. ED490160.

Rhetoric: 2010. In *Merriam-Webster Online Dictionary*.Retrieved October 13, 2010 from http://www.merriam-webster.com/dictionary/rhetoric

Lombard, M., & Ditton, T. (1997). At the heart of it all: The concept of presence. *Journal of Computer Mediated-Communication* [On-line], *3 (2)*. Available: http://www.ascusc.org/jcmc/vol3/issue2/lombard.html

Fung, A. (2006). *Empowered participation: Reinventing urban democracy.* Princeton, NJ: Princeton University Press.

McConnell, D. (2006) *E-Learning Groups and Communities*. Maidenhead, Open University Press.

Nardi, B. (2005). Beyond bandwidth: Dimensions of connection in interpersonal interaction. *The Journal of computer supported cooperative work* 14: 91-130.

Steeples, C, Jones CR & Goodyear, P (2002) Beyond e-learning: the future for networked learning. In C

Steeples & C Jones (Eds) *Networked learning: perspectives and issues*. London: Springer

Suthers, D. (2001). Collaborative representations: Supporting face to face and online knowledge-building discourse. *Proceedings of the 34th Hawai`i International Conference on the System Sciences* (HICSS-34), January 3-6, 2001, Maui, Hawai`i: Institute of Electrical and Electronics Engineers.

Thatchenkery, T., & Metzker, C. (2006). *Appreciative intelligence: Seeing the mighty oak in the acorn.* San Francisco: Berrett-Koehler.

Tomadaki, E., Quick, K., & Scott, P. (2008). Videoconferencing in Open Learning. *Journal Of Interactive Media In Education, 2008*(1). Retrieved November 3, 2010, from http://jime.open.ac.uk/article/2008-8/341

Reed, J., & Woodruff, M. (1995). An introduction to using videoconferencing technology for teaching, The Distance Educator Newsletter, Fall, Scott, T D and Pitcher, N (1997) SUMSMAN: Scottish, from: http://www.kn.pacbell.com/wired/vidconf/Using.html

Ryan, S., Scott, B., Freeman, H., & Patel, D. (2000). *The virtual university.* London: Kogan Page.

Saw, ,K.G, Majid, O., Abdul Ghani, N., Atan, H.,Idrus, R. M., Rahman, Z. A.,Tan, K. E. (2007). The videoconferencing learning environment: technology, interaction and learning intersect. *British Journal of Educational Technology*, *39*(3), Retrieved from http://dx.doi.org/10.1111/j.1467-8535.2007.00736.x doi: .doi.org/10.1111/j.1467-8535.2007.00736.x DO - 10.1111/j.1467-8535.2007.00736.x

Short, J., Williams, E. & Christie, B. (1976). *The social psychology of telecommunications*. London: John Wiley & Sons.

Strijbos J.W., Martens R.L. and Jochems W.M.G. (2003). Designing for Interaction: Six Steps to Computer-supported Group-based Learning. *Computers and Education*

Smyth, R. (2005). Broadband videoconferencing as a tool for learner-centred distance learning in highereducation. *British Journal of Educational Technology, 36*(5), 805–820.

Thomson, A., & Gutmann, D. (2004). *Why deliberative democracy.* Retrieved May 8, 2008, fromhttp://press.princeton.edu/chapters/s7869.pdf.

Townsend, A.M., Hendrickson, A.R. and DeMarie, S.M. (2002). Meeting the Virtual Work Imperative. *Communications of the Association for Computing Machinery*, 45(1)

Sedgwick, M. & Spiers, J. (2009). The use of videoconferencing as a medium for the qualitative interview. *International Journal of Qualitative Methods,* 8(1). Available from https://ejournals.library.ualberta.ca/index.php/IJQM/index

Zeichner, K. (2001). Educational Action Research. In P. Reason & H. Bradbury(Eds.), *Handbook of action Research: Participative inquiry and practice* (pp.273-282). London: SAGE publications.

.

SESSION

TEACHING METHODS AND RELATED ISSUES

Chair(s)

TBA

Using Templates to Introduce Time Efficiency Analysis in an Algorithms Course

Irena Pevac

Department of Computer Science
Central Connecticut State University, New Britain, CT, USA

Abstract: *We propose introducing algorithm time efficiency analysis by utilizing a set of template examples that illustrate the following seven basic time performance complexity categories: log n, n, n log n, n^2, n^3, a^n, and n!. The complexity category of some function f(n) is the set of all functions that have the same asymptotic growth as f(n), which is denoted Θ(f(n)). The template examples are simple, their time efficiency analysis is not mathematically intensive, and they correspond to various algorithms. Both iterative and recursive algorithm templates are discussed. In addition, we provide template modifications that do not affect asymptotic growth. Modified templates remain in the same complexity category. Finally, we list several problems for which algorithm time performance belongs to a given complexity category. We argue that the approach of using templates and their transformations not only complements the introduction to time efficiency analysis, but also can improve understanding of this traditionally difficult topic in an Algorithms course. Its advantage is that it sheds some light as to which part or steps of the algorithm affect its run time the most. The template approach also provides flexibility for reaching the appropriate level of in depth discussion suitable for particular learner's background.*

Keywords: Algorithms, Time Performance Analysis, Teaching

1 Introduction

Algorithm time efficiency is an important topic in an algorithms course. This paper discusses the mathematical approach to analyze time efficiency which is independent of the implementation and machine performance details and is used in algorithm course textbooks [1] to [8], [10], and [11].

Most textbooks introduce formal analysis with O, o, Θ, ω, and Ω, and illustrate growth rates functions: log n, n, n log n, n^2, n^3, a^n, and n! by displaying their values for different values of n, and by displaying pictures of plotted functions together to provide comparison. The introduction is then typically followed by detailed time performance analysis of several algorithms.

The students in an algorithms course often experience difficulty understanding formal analysis of mathematically involved advanced algorithms from various domains and design types.

We believe that: 1) learning templates for each of the seven basic complexity categories; 2) learning which modifications of templates produce results that still remain in the same complexity category; and 3) practicing time efficiency analysis of simple algorithms for each complexity category should be done before starting with mathematically intensive analysis of algorithms that are difficult to grasp. This also smooths the transition into studying how to group algorithms by their design type and their domain. We strongly believe that this approach will improve learning the traditionally difficult topic of time efficiency analysis.

2 Templates for the seven basic complexity categories

In the next subsections we discuss several templates used in many simple problems to analyze algorithms that belong to the following seven basic complexity categories: log n, n, n log n, n^2, n^3, a^n, and n!. Both iterative and recursive algorithm templates are discussed.

Java based pseudo-code is used to write templates.

Next, for each template, we list possible modifications that will not change its asymptotic time performance. We omit the illustration of each type of modification with concrete values and also omit proof that it does not change asymptotic growth. In the classroom, an instructor can provide the proof for one or two complexity categories and students can be divided into groups to prove the rest.

At the end, for each of the seven complexity categories, we list several examples of problems that can be implemented by using one of the templates to produce an algorithm that belongs to that particular complexity category. More algorithm examples for various complexity categories can be found in [9]. In the classroom students could be asked to provide additional examples for each category. Again, work could be distributed among several student groups.

2.1 Templates for logarithmic run time

Template 1

```
for (i=1; i<=n; i=i*2)
     do one basic operation
```

Variations for template 1 include one or more of the following: 1) control variable i can be initialized to start at any nonnegative constant instead of starting at 1; 2) performing one basic operation in the loop can be replaced with performing any constant number of basic operations; and 3) control variable i can be multiplied by any other positive constant c in place of 2.

Template 2

```
for (i=n; i>=1; i=i/2)
    do one basic operation
```

Variations for template 2 include one or more of the following: 1) condition to stop looping can have control variable i greater than or equal to a nonnegative constant; 2) performing one basic operation in the loop can be replaced with performing any constant number of basic operations; and 3) control variable i can be divided by any positive constant c instead of 2.

Template 3

```
LogRec(int n)
    if (n==1)
          do one basic operation
    else
         { do one basic operation
           LogRec(n/2) }
```

Variations for template 3 include one or more of the following: 1) base case can be when variable n has one or more of given initial constant values; 2) performing one basic operation can be replaced with performing any constant number of basic operations either in the base case or in the recursive step part; and finally, 3) a recursive call can be made with variable n reduced by any positive constant factor c instead of 2. Obviously, it must be provided that base case is reached.

The following simple algorithms with logarithmic time performance can be used to illustrate the above template examples: 1) determine the number of digits in binary representation of a given decimal number n; 2) determine the number of digits of a given integer number n; 3) convert base ten (decimal) number n into base 2 (or base 3, or base 8, or any other constant); 4) calculate a raised to the power of n based on formula $a^n = 1$ if n=1, $a^n = a^{n/2} * a^{n/2}$ if n>1, and n is even, and $a^n = a * a^{n/2} * a^{n/2}$, otherwise; 5) print recursively letter "A" $\log_b n$ times; 6) for given integer number n return the string of digits representing number n in base 3; 7) binary search for worst case (unsuccessful search); and 8) display the digits of a given positive integer n in reverse order.

2.2 Templates for linear run time

Template 4

```
for (i=1; i<=n; i=i+1)
    do one basic operation
```

Variations for template 4 include one or more of the following: 1) control variable i can be initialized to start at any nonnegative constant; 2) performing one basic operation in the loop can be replaced with performing any constant number of basic operations; and 3) control variable i can be increased by any positive constant c instead of 1.

Template 5

```
for (i=n; i>=1; i=i-1)
   do one basic operation
```

Variations for template 5 include one or more of the following: 1) the initial value for variable i can be n + constant (positive or negative) instead of n; 2) condition to stop looping can have control variable i greater than or eaqual to any positive constant; 3) performing one basic operation in the loop can be replaced with performing any constant number of basic operations; and 4) control variable i can be decreased by any positive constant c instead of 1.

Template 6

```
LinearRec(int n)
    if n=1
       do one basic operation
    else
      { do one basic operation
        LinearRec(n-1) }
```

Variations for template 6 include one or more of the following: 1) base case can be when variable n has one or more of given constant values; 2) performing one basic operation can be replaced with performing a constant number of basic operations in the base case or in the recursive step part; and finally, 3) recursive call can be made with variable n decreased by any positive constant c instead 1, provided that base case is reached..

Template 7

```
LinearRec(int n)
    if n>1
      { do one basic operation
        LinearRec(n/2)
        LinearRec(n/2)  }
```

Variations for template 7 include one or more of the following: 1) base case can be when variable n has one or more given constant values; 2) performing one basic operation can be replaced with performing any constant number of basic operations in the base case or in the recursive step part; 3) the order of performing one or more basic operations and making two recursive calls to sub-problems can be any permutation of those three steps; 4) several basic operations can be done before, in between, or after the two recursive calls; and finally, 5) recursive calls can be made with variable n reduced by another positive constant factor c instead of 2.

Template 8

```
LinearRec(int n)
    if n>1
         { do one basic operation
           LinearRec(n/3)
           LinearRec(n/3)
           LinearRec(n/3)  }
```

Variations for template 7 include one or more of the following: 1) base case can be when variable n has one or more of given constant values; 2) performing one basic operation can be replaced with performing any constant number of basic operations in the base case or in the recursive step part before, after, or in between any two of the three recursive calls to sub-problems; 3) a constant number k of recursive calls can be made instead of three calls, where each recursive sub-problem has variable n reduced by factor k .

The following simple algorithms with linear time performance can be used to illustrate the above discussed template examples 4-7: 1) determine the sum of first n numbers; 2) determine the sum of first n squares; 3) determine the sum of first n cubes; 4) determine the sum of array components; 5) determine the harmonic sum; 6) determine the largest array component; 7) determine the index of the largest array component; 8) perform a linear search for a given key in the array with n components; 9) calculate a raised to the power of n based on the formula $a^n = 1$ if n=1, and $a^n = a^{n/2} * a^{n/2}$ if n>1 and nis even, and $a^n = a * a^{n/2} * a^{n/2}$ otherwise, where two identical recursive calls to sub-problems of size n/2 are made in the code; and 10) print letter "A" n times.

2.3 Templates for n - log n run time

Template 9

```
for (i=1; i<=n; i=i+1)
   for (j=1; j<=n; j=j*2)
      do one basic operation
```

Variations for template 9 include one or more of the following: 1) control variable i of the outer loop can be initialized to start at any nonnegative constant instead of starting at 1; 2) performing one basic operation in the inner loop can be replaced with performing any constant number of basic operations; 3) control variable i can be increased by any positive constant instead of being increased by 1; 4) control variable j can be initialized to any positive constant; and 5) control variable j can be multiplied by another positive constant c instead of being multiplied by 2.

Finally, since the two loops are not correlated they can be interchanged without affecting the time performance.

Template 10

```
N_Log_N_Rec(int n)
   if (n>1)
      { N_Log_N_Rec( n/2 )
        N_Log_N_Rec( n/2 )
        Perform n basic operations }
```

Variations for template 10 that do not affect asymptotic time performance include one or more of the following: base case might be one of more values for n that are constant. For such base case values any constant number of basic operations can be performed. During the recursive step the basic operation can be performed an+b times where a and b are constants. Those basic operations can be performed before, after or in between the two recursive calls of size n/2.

Template 11

```
N_Log_N_Rec(int n)
   if (n>1)
      { N_Log_N_Rec( n/4 )
        N_Log_N_Rec( n/4 )
        N_Log_N_Rec( n/4 )
        N_Log_N_Rec( n/4 )
        Perform n basic operations }
```

Variations for template 11 that do not affect asymptotic time performance include one or more of the following: 1) base case might be one of more values for n that are constant. For such base case values any constant number of basic operations can be performed; 2) during the recursive step the basic operation can be performed an+b times where a and b are constants. Those basic operations can be performed before, after or in between any two of the four recursive calls of size n/4.

The following algorithms with n - log n time performance can be used to illustrate the above discussed template examples 9 - 11: 1) draw letter "A" n log n times; 2) merge sort; 3) heap sort; and 4) draw the following squares using recursive algorithm. Base case is when side length is less than or equal to a given constant. No basic operation is performed in that case. At the highest level of recursion, one rectangle centered at x,y with side length equal to n is drawn. In addition, four recursive calls are made with center coordinates shifted 2n in each of the following four directions: 1) up, left, 2) up, right, 3) down, left, and 4) down, right. Each of the four recursive calls has size n reduced by factor 4.

2.4 Templates for quadratic run time

Template 12

```
for (i=1; i<=n; i=i+1)
   for (j=1; j<=n; j=j+1)
       do one basic operation
```

Template 13

```
for (i=1; i<=n; i=i+1)
   for (j=i; j<=n; j=j+1)
     do one basic operation
```

Template 14

```
for (i=1; i<=n; i=i+1)
   for (j=1; j<=i; j=j+1)
      do one basic operation
```

All variations described for a single loop for linear time are applicable for each of the two nested loops in template examples 12-14 .

Template 15

```
QuadraticRec(int n)
   if (n>0)
     { do n basic operations
       QuadraticRec( n-1) }
```

Variations to template 15 that preserve quadratic run time include: 1) base case and/or recursive step where several constant number of basic operations is performed; 2) a recursive call to a sub-problem with size decreased by constant value instead of n-1; 3) the order of recursive call to a sub-problem, and performing one, or performing some constant number of basic operations, can be interchanged.

Template 16

```
QuadraticRec(int n)
   if (n>1)
  {   QuadraticRec( n/2 )
      QuadraticRec( n/2 )
      QuadraticRec( n/2 )
      QuadraticRec( n/2 )
       Do constant number of basic operations }
```

The following simple algorithms with quadratic time performance can be used to illustrate the above discussed template examples 12 - 16: 1) add all components in a given n x n matrix; 2) determine average of all components in a two dimensional array of size n x n; 3) determine the largest of all components in a two dimensional array of size n x n; 4) selection sort; 5) insertion sort; 6) bubble sort; 7) draw given letter "A" n squared times; 8) draw the squares using the following recursive algorithm: base case is when side length is less than or equal to a given constant. No basic operation is performed for the base case. At highest level of recursion, one rectangle centered at x,y with side length equal to n is drawn. In addition, four recursive calls are made with the center coordinates shifted 2n in each of the following four directions: 1) up, left, 2) up, right, 3) down, left, and 4) down, right. Each of the four recursive calls has size n reduced by factor 2; 9) draw the squares using the following recursive algorithm. Base case is when side length is less than or equal to a given constant. One basic operation is performed in that case. At the highest level of recursion, one rectangle is drawn centered at x,y with side length equal to a given constant. In addition, four recursive calls are made with center coordinates shifted 2n in each of the following four directions: 1) up, left, 2) up, right, 3) down, left, and 4) down, right. Each of the four recursive calls has size n reduced by factor 2.

2.5 Templates for cubic run time

Template 17

```
for (i=1; i<=n; i=i+1)
  for (j=1; j<=n; j=j+1)
     for (k=1; k<=n; k=k+1)
        do one basic operation
```

Template 18

```
for (i=1; i<=n; i=i+1)
  for (j=1; j<=i; j=j+1)
     for (k=1; k<=n; k=k+1)
        do one basic operation
```

Variations are analogous to the variations specified for iterative quadratic templates.

Template 19

```
CubicRec(int n)
   if (n>1)
  {    do n squared basic operations
       CubicRec(n-1)}
```

Variations are analogous to the variations for the LinerRec example.

Template 20

```
CubRec(int n)
  if (n>1)
     { do one basic operation
       CubRec(n/2)
       CubRec(n/2)
       CubRec(n/2)
       CubRec(n/2)
       CubRec(n/2)
       CubRec(n/2)
       CubRec(n/2)
       CubRec(n/2)}
```

Variations to template 20 include base case for n equal to a given constant or n less than given constant and performing up to n*n basic operations. In recursive step, performing one basic operation can be replaced by performing a linear or quadratic number of basic operations.

The following simple algorithms with cubic time performance can be used to illustrate the above discussed template examples 17 - 19: 1) add all components in a given n x n x n matrix; 2) determine the average of all components in a three dimmensional array of size n x n x n; 3) determine the largest of all components in a three dimmensional array of size n x n x n; 4) determine the lengths of the shortest paths between all pairs of vertices in a graph represented with adacency matrix (use Floyd algorithm); 5) print letter the "A" n^3 times; 6) determine the product of two n x n matrices using a brute force algorithm; 7) determine if a given three sets of cardinality n are disjoint; 8) apply Gaussian elimination to transform n x n matrix into upper triangular form; and 9) use Floyd algorithm for shortest paths between all pairs of vertices in a graph represented with adjacency matrix.

2.6 Templates for exponential run time

Template 21

```
IterativeExponential(int n)
  limit=1
  for (i=1; i<=n; i++)
    {    for(j=1; j<= limit; j++)
            do one basic operation
         limit = limit * 2}
```

Variations include: 1) starting loop variables from given constants instead of 1; or 2) ending the outer loop at n – const; 3) we can replace the statement limit= limit*2 with the statement limit= limit*k where k is a constant that is greater than 2. The run time will be a faster growing class but it will still be exponential k^n.

```
Template 22

ExponentialRec(int n)
  if (n=0)
      do one basic operation
  else
 {     ExponentialRec( n-1 )
       ExponentialRec( n-1 )}
```

Variations include base case at n equal to or less than a given constant and performing a constant number of basic operations at base case or anywhere during the recursive step.

2.7 Templates for n factorial run time

```
Template 23

FactorialRec(int n)
   if (n=1)
     do one basic operation
   else
     for (int i=1; i<=n; i++)
        FactorialRec(n-1)
```

The following algorithms with n factorial time performance can be used to illustrate the template example 23: 1) print all permutations for given n; 2) print all paths for traveling salesman (exsaustive search brute force algorithm); and 3) print letter "A" n factorial times.

3 Future work

The above described approach could be used to design a learning tool similar to ESRATEA (Pevac, 2010) intended for learning and practicing time performance analysis of all types of algorithms. Currently, ESRATEA is designed for recursive algorithms only. Such learning tool could be used as a supplemental course material for Algorithms courses using several textbooks.

References

[1] Aho, A. V., Ullman, J. D. (1995). *Foundations of Computer Science* (2nd ed.). Computer Science Press.

[2] Baase, S., Van Gelder, A. (2000). *Computer Algorithms* (3rd ed.). Addison Wesley Longman.

[3] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Clifford, S. (2007). *Introduction to Algorithms* (2nd ed.). MIT Press; McGraw-Hill Book Company.

[4] Goodrich M., Tamassia R. (2002). *Algorithm Design: Foundations, Analysis, and Internet Examples* John Wiley & Sons.

[5] Johnsonbaugh, R., Schaefer, M. *Algorithms* (2004). Pearson.

[6] Levitin, A. (2007). *The Design & Analysis of Algorithms* (2nd ed.). Pearson- Addison Wesley.

[7] McConnell, J. J. (2008). *Analysis of Algorithms* (2nd ed.). Jones and Bartlett Publishers.

[8] Neapolitan, R., Naimipour K. (2004). *Foundations of Algorithms Using Java Pseudocode* Jones and Bartlett.

[9] Pevac, I. ESRATEA Educational Software for Recursive Algorithm Time Efficiency Analysis *WORLDCOMP'10 Proc. of the 2010 Int. Conf. on Frontiers in Education: Computer Science & Computer Engineering*, (Ed. Arabnia, Clincy), Las Vegas, NV, SCREA Press, 2010, pages 367-373.

[10] Sedgewick, R., Wayne, K. (2011) *Algorithms* (4th ed.). Addison Wesley.

[11] Shaffer C. (1998) A Practical Introduction to Data Structures and Algorithm Analysis Prentice Hall.

The Teaching Capability Maturity Model for Teachers in Higher Education: a Preliminary Study

C.Y. Chen[1], C.Y. Kuo[2] , P.C. Chen[3]
[1]Dept. of Information Management, National Central University,
300 Jhong-Da Rd., Jhong-Li, Taiwan
[2]Dept. of Information Management, Chang-Gung University
259 Wen-Hua 1st Rd., Tao-Yuan, Taiwan
[3]Dept. of Information Management, National Taiwan University,
No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan

Abstract – *To address teaching quality in higher education, in addition to the organizational focus and the momentary evaluation methods (e.g. students' or peer-based teaching evaluations), emphasis should also be put on teachers and their teaching processes. In software industry, there is a model called Capability Maturity Model Integration (CMM/CMMI). CMM/CMMI collects a set of software processes for a development team to follow. It also defines several levels of process institutionalization for evaluating the stability of the software team. In this research, we attempt to apply the CMM/CMMI concepts to higher education, and develop a Teaching Capability Maturity Model (T-CMM) for tertiary teachers to manage and improve teaching quality. The entire research consists of two parts: (1) the applicability of CMM/CMMI to teaching quality in higher education and (2) the development of the T-CMM. This paper serves as the first part; it introduces CMM/CMMI, and then identifies several issues concerning the application. Finally, this paper outlines the future development of such an applied model.*

Keywords: higher education, teaching process focus, capability and maturity model

1 Introduction

Teaching quality has been the main focus of universities and colleges in their continuing quest for higher education. Researchers in educational or quality related fields have been working on methods and standards to address teaching quality. Most of these methods and standards, e.g. ISO/IWA2, AACSB and ABET, etc., focus on the organizational level. For example, the main essence of ISO/IWA2 is a criterion to demand/regulate the quality of education, which emphasizes what the institution should do to ensure education and teaching quality (ISO/IWA2, 2003). However, it is teacher who plays the key role in delivering the quality of teaching, and this is particularly true in higher education where teachers and faculties tend to be self-governed and independent (George, 1997; Seldin, 1997). If education also agrees with the concept of quality management, then one way to address teaching quality in higher education should start from the quality contributors, that is, teachers, rather than solely focusing on the organization as a whole (Horine *et al*., 1993).

In addition to the focus on teachers, process plays another critical role in teaching quality (Grant *et al*., 2004; Louden, 2000). Existing evaluations on a teacher's teaching quality, such as students' feedback questionnaires or onsite peer evaluation, are largely dependent on students' subjective feeling about the teacher or how nice the teacher can be, or the momentary performance during the peer visit (Cramer and Alexitch, 2000; Marsh and Roche, 1997; McKeachie, 1997). These result-oriented methods are sometimes too judgmental or partial, neglecting other efforts a teacher may have made during the processes of the entire course.

When the focus is put on the teacher and the teaching processes, this paper argues that teaching quality can be addressed and managed by recognizing and improving the processes, as well as by maintaining the performance of the processes for a teacher from course to course. In other words, teaching quality should comprehensively cover the processes of a teacher from planning a course, instructing and monitoring the course, as well as the conveyance of lessons learned from this course over to the next session.

In software industry, there is a process reference model called Capability Maturity Model Integration (CMM/CMMI) for standardizing and stabilizing software development. CMMI collects relevant processes of software development for a software team to follow. It also defines several levels of process institutionalization for evaluating the stability of the software team. In our research, we attempt to apply CMM/CMMI in higher education and establish the Teaching Capability Maturity Model (T-CMM) for tertiary teachers to manage and improve their teaching quality. However, because higher education and software development are two different domains, the development of T-CMM requires first to consider and identify the differences. In this regard, the entire research consists of two parts: (1) the applicability of

CMM/CMMI to teaching quality in higher education and (2) the development of the T-CMM. This paper serves as the first part; it introduces CMM/CMMI, and then identifies several issues concerning the application.

2 Capability Maturity Model Integration (CMM/CMMI)

In his book *Quality is Free*, Crosby introduced the concept of "quality staging" in required processes implementation (Paulk et al., 1993). Based on this concept, and by applying total quality management to project oriented software development, SEI (Software Engineering Institute at Carnegie Mellon University) (2010) established a comprehensive process model called Capability Maturity Model (CMM), and a newer, integrated version called CMMI, to address software development quality in terms of: (1) the maturity level (namely, the overall and systematic performance) or (2) the capability level (that is, when focusing on a particular process).

3.1. Process areas, capability/maturity levels & institutionalization design in CMM/CMMI

To be comprehensive, the CMM/CMMI framework comprises a number of core process areas (PA) that cover the full range of building blocks from engineering practices, such as: coding, testing, systems integration, etc, to managerial practices, such as project and process management. For example, Project Planning, Project Monitoring and Control, etc. are project management related PA; Requirement Development, Product Integration, etc. are engineering related PA. Each PA has specific goals (SGs) and related specific practices (SPs, that is, best practices) that project developers can use to implement these specific practices to meet specific goals. To institutionalize these implementation results, CMMI further defines generic goals (GGs) and generic practices (GPs), and classifies them into different institutionalized stages or levels, that is, maturity levels (ML) or capability levels (CL).

We use the following two tables to further describe the feature designs of the CMM/CMMI. Table 1 presents the definitions of the capability and maturity levels in CMM/CMMI. Table 2 on the next page is the summary of the institutionalization contents in CMM/CMMI.

Table 1: The maturity and capability levels in CMMI

Maturity Level (ML)	Capability Level (CL)
1. ***Initial***：the processes are chaotic in order to successfully complete the software project	**0.** ***Incomplete***: cannot satisfy all the specific goals (SG) in a teaching process area (PA).
2. ***Managed***—At this stage, basic project management PAs are managed and controlled. Similar projects would have a similar performance.	**1.** ***Performed***: satisfy all the SGs in a PA **2.** ***Managed***: the performance of a PA is similar in similar courses
3. ***Defined***— At this stage, PAs are characterized and are able to be tailored. Process improvement is established and sustained.	**3.** ***Defined***: a PA is characterized and is able to be tailored. Process improvement is established and standardized for improving the process area.
4. ***Quantitatively Managed***—At this stage, software processes can be quantitatively managed by quantitative objectives.	**4.** ***Quantitatively Managed***: a PA is quantitatively managed based on established quantitative objectives.
5. ***Optimizing***—At this stage all PAs are continually improved and sustained at their optimal situation.	**5.** ***Optimizing***: a PA is continually improved and sustained their optimal situation.

Table 2: The concept of generic goals and practices in CMMI

GG1 To achieve specific goals *GP1.1 Perform the specific practices:* Organizations are asked to implement the specific goals of a PA.
GG2 To institutionalize the managed process *GP2.1 Establish policies:* Organizations should establish policies for performing a PA. *GP2.2 Plan the process:* Software team should plan how to perform a PA *GP2.3 Provide resources:* Organizations should provide adequate resources for performing the processes. *GP2.4 Assign responsibilities:* Responsibilities of the stakeholder 0of the project should clearly defined.. *GP2.5 Train people:* To ensure proper training is placed for performing a PA *GP2.6 Manage configurations:* The contents and work products produced in a PA should be managed. *GP2.7 Identify & involve relevant stakeholders:* Relevant stakeholders should be identified. *GP2.8 Monitor the process:* The performance and issues when performing a PA should be monitored. *GP2.9 Objectively evaluate the adherence:* Organizations should objectively evaluate how well the relevant stakeholders follow the processes.

GP2.8 Communicate with the management: The technical and managerial results of performing process should inform the management.
GG3 To institutionalize the defined process *GP3.1 Establish defined processes:* PAs are standardized for a normal situation and are characterized for mgt. needs. *GP3.2 Collect improvement info:* Organizations collect and handle the improvement information for a PA.
GG4 To institutionalize the quantitative managed process *GP4.1 Establish quantitative objectives:* For each process area one should establish quantitative performance objectives for management needs. *GP4.2 Stabilize sub processes:* A process can be decomposed into sub, quantifiable process elements.
GG5 To institutionalize the optimizing process *GP5.1 Ensure continual process improvement:* Organizations should continually improve a PA, including learning and using more innovative approaches in improving the performance of the PA *GP5.2 Resolve the root causes of problems:* Organizations should continually identify and resolve common root causes for all the PAs and prevent them from occurring.

2.2. Equivalent staging

In CMM/CMMI, an organization can either choose to implement a group of teaching process areas at a maturity stage, or to focus on a specific PA for achieving a capability level of that PA. To compare the organizations that have different representations (i.e. ML or CL), CMM/CMMI provides a mapping function termed equivalent staging that provides organizations the flexibility in using the model. In the concept, a ML accomplishment means that all the process areas on that maturity stage also meet the same level of capability.

For example, CMMI (version 1.3) ML-2 comprises seven process areas, i.e. Requirement Management (REQM), Project Planning (PP), Project Monitoring and Control (PMC), Configuration Management (CM), Product and Process Quality Assurance (PPQA), Measurement and Analysis (MA) and Supplier Agreement Management (SAM). When an organization reaches the capability level 2 for each of the seven process areas, the organization also obtains a maturity level 2 (ML-2) rating since these process areas constitute the maturity level-2 stage.

3 Applying CMMI to teaching quality

The concepts of maturity and capability have been applied to various process-oriented activities in other domains. Recently, CMM/CMMI has also received an attention in the educational field. For example, White *et al.* (2003) and Daigle *et al.* (2003) applied the CMMI concepts to the "IS 2002 Model Curriculum (ACM, 2002)", a recognized model used for improving the IS (Information System) curriculum. They utilized CMMI to enable institutions to use the curriculum model and certification exams to attain the desired maturity level of organizational performance. Subsequently, a maturity model for engineering education (CEMM), which was proposed by Lutteroth *et al.* (2007), showed the applicability and feasibility of CMMI to engineering education institutes. The same idea of applying the CMMI's maturity concept to higher education is found in Dounos and Bohoris' (2007) conceptual research paper. Table 3 lists the CMM/CMMI applications, including the educational domain.

While these studies described in above paragraph may initially indicate the applicability of CMM/CMMI in higher education, they only outline the maturity levels without any actual model implementation. The CMM/CMMI applications in these studies do not elaborate nor provide any details of the "what-to-do" (i.e. specific goals and specific practices) contents regarding these building blocks. Furthermore, these CMM/CMMI applications focus mostly on organizations as a whole. By contrast, T-CMM focuses on teachers, not the organizations. Figure 1 below illustrates the draft application framework of T-CMM.

While in its attempt to use the CMM/CMMI concepts for the quality of teaching, T-CMM seems to specialize in substantial ways due to different domains, and the applicability deserves further examination. In examining the applicability, this paper identifies four applicability arguments: the nature of the development, the deliverer of quality, ongoing quality improvement, and the sustainability of quality achievement. They are described in the next section.

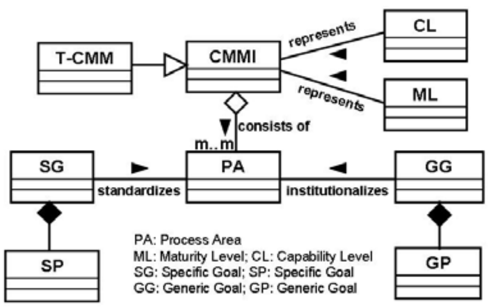


Figure 1: The draft application framework

Table 3：Applying the concept of CMM/CMMI in other domains, including education

Applications	Usage
Relationship management maturity model (Martin *et al*.,2004)	The communication between IT department and the rest of the organization has been a challenge. Martin et al. applied the maturity concept and developed a relationship management maturity framework for managing and improving the complicated service relationships.
Financial management capability model (McRoberts and Sloan,1998)	The capability model provides a tool that an auditor or government organization can use to: determine its financial management requirements based on the nature, complexity and associated risks of its operations; assess its financial management capabilities against established requirements.
e-Business capability maturity model (Chen and Yu, 2008)	Ad-hoc EB implementations conducted by organizations have different levels and scopes, but they are all regarded as e-business achievement. Chen and Yu developed EB-CMM as a reference model not only for organizations who want to implement their e-business environment; but also to provide a communication protocol as players and partners in supply chains to establish a profound and win-win e-business collaboration environment.
Medical security capability model (Williams, 2008)	Williams applied the capability concept in the field of medical information security and proposed a medical information security capability framework.
Data Warehousing Process Maturity (Sen *et al*., 2006)	Data warehousing, like software development, is a process, which can be expressed in terms of components such as artifacts and workflows. The authors draw upon the concepts underlying CMM to define different maturity levels for a data warehousing process (DWP).
Cognizant Enterprise Maturity Model (Harigopal and Satyadas, 2001)	A cognizant enterprise is defined as an enterprise capable leveraging knowledge and learning to introduce smartness and adaptability into business processes and the operational systems that support these business processes. Cognizant enterprise maturity model (CEMM) [16] provides the roadmap for organizations to qualify and achieve these capabilities
A maturity model for construction (Hutchinson and Finnemore, 1999)	A process improvement model, Standardized process improvement for construction enterprises (SPICE), applying CMM for the construction industry.
Maturity model for computing education (Lutteroth *et al*., 2007)	The conceptual model (CEMM) is designed to be used in rating educational organizations according to their competence to deliver high quality education on a five-level scale. Only maturity levels are delivered in the work.
A maturity model for online course design (Neuhaser, 2004)	The author analyzes the online course design and suggests the concept of maturity process, which is technological oriented.

4 Applicability issues

4.1. Nature of development

According to the SEI (2010), CMM/CMMI is designed for software development, where the goals for each development are highly customized. Therefore, the development results can rarely be entirely standardized and duplicated. This is because software development is dynamic in nature, and there are no two projects have the exactly same contents (SEI, 2010). In spite of this, CMM/CMMI suggests that software organizations establish common practices of their projects, and leave proper abstraction regarding the contents of the common practices. By doing so, individual projects are able to override or implement the practices to best suit unique project characteristics and various development situations.

As mentioned earlier in this paper, teachers in higher education are domain independent and receive respect regarding how they teach their courses. In other words, higher education emphasizes exploring knowledge in various domains, teachers in higher educational organizations often attract respect and are free to carve their own niches in developing and conducting their course teaching (Seidin, 1997; George, 1997; Barnett, 1992). So here comes a question: Do there exist common practices for the tertiary teachers?

This is a critical issue, since the requirements of standardization in CMM/CMMI is constructed based on the common practices of software development. As for the dynamics of a course teaching, regardless the controversy of whether students can be regarded as products, what we observe is that, teachers often teacher identical courses for a number of times. Under this situation, are the teaching contents of a repeated course identical? Besides, It is known that students' backgrounds (heterogeneity of "raw materials"?) are different and learning results (that is, the "development" result) often vary during the delivery of each term's course. In spite of this, should teachers spend extra effort on planning a course, especially when the course has been taught for numerous times and the teacher is very experienced? If so, what are the factors that a teacher needs to consider when planning a repeated course? CMMI requires the establishment of tailoring guidlines for projects to plan their processes. It that possible for teachers to establish such tailoring guidelines/rules for themselves to follow during the time of planning of each course?

4.2. Deliverer of quality

In CMMI, the deliverer of quality refers to the development teams and the software organizations as a whole. Therefore, the actors of many specific and generic practices (that is, SPs, GPs) are designed to be software organizations and teams. Because of this, CMMI does not attribute to individuals when the performance of the processes falls behind the expectation.

In higher education, however, the main deliverer of teaching quality is individual teacher. In this instance, assessing teaching quality by using the collective organization as a subject may not be appropriate, since a teacher's teaching quality cannot stand for other teachers, nor can it be duplicated. Moreover, because higher education respects domain independence and teaching individuality, a process reference model of teaching quality should provide only the abstraction of teaching practices and leave the implementation details (which are regarded as respective characteristics) to teachers. These quality deliverer issues should also be addressed when applying CMMI to teaching quality in higher education.

4.3. Continuity of quality improvement:

The capability and maturity design in CMM/CMMI provides a roadmap regarding gradual and continual improvement for better quality achievements. In other words, CMM/CMMI does not require software teams to achieve these levels at once, but requires them to keep up the good work. According to SCAMPI (Standard CMMI Appraisal Method for Process Improvement) (SEI, 2006), maturity or capability appraisals are performed on a periodical basis. The purpose of the appraisal is to ensure that the improvement (climbing to a higher maturity or capability level) has been made from previous assessment, as well as to ensure that the implemented practices in the previous assessment have been maintained.

It is true that in higher education, teachers may not be born to teach. Many of them have not even received any training prior to teaching in universities. Under these circumstances, the capability maturity concept seems to offer teachers a roadmap for attaining teaching excellence. However, to experienced teachers or professors, should they also continually improve their teaching methods? What is the motivation for experienced teachers and professors to take effort for self improvement?

Besides, existing teaching quality evaluation tools, such as students' feedback questionnaires or onsite peer evaluation, also provide some useful information regarding continual improvement. Here raises an issue: what is the functional relationship between teaching capability/maturity and these existing evaluation methods? Or is it just another annoying thing that even confines the professors' teaching development? In the discussion of executing teaching quality evaluation, existing CMM-based teaching reference models, such as (Lutteroth *et al*., 2007; Dounos and Bohoris, 2007; White *et al*; 2003; Daigle *et al*.; 2003), have no problems as they focus on educational institutes. Since T-CMM focuses on individual teachers, how is a teaching capability and maturity assessment carried out? In particular who performs the assessment and how often to perform the assessment? These issues need to be addressed in T-CMM.

4.4. Organizational vs. individual's sustainability of quality

While both CMM/CMMI and teaching quality emphasize on process, CMM/CMMI focuses on the repetition of the organization's established practices. In order for teaching quality to be repeatable, established teaching practices would also need to be sustained. However, sustainability of teaching excellence should focus more on teachers, who are to maintain their respective characteristics and teaching excellence.

For example, to institutionalize the implementation results, the GP2.3 (provide rfesources) in Table 2 suggests that CMM/CMMI implementers (organizations) need to provide sufficient resources, and the appraisal examines the organizations to see if they have met this requirement. In teaching quality, in order for teachers to sustain teaching quality, required resources that support excellent teaching quality need to be ensured as well. However, the implementer of teaching capability maturity is individual teachers, who may not be able to provide all of the required resources (e.g. classrooms, projectors, air conditioners, etc.) for teaching. In this regard, the framework of CMM/CMMI's institutionalization design may be applicable to teaching quality; yet some of the contents may not be applicable, since achievement in teaching quality refers to personal excellence that is retained in the individual teachers, not the organizations.

5 What is next

When the focus is put on teachers (*NOT* the educational institutes) and their teaching processes, the concepts of CMM/CMMI, including process area design, capability/maturity level design, and institutionalization design, seem to be useful for teachers in managing and improving their teaching quality. In the follow-up research, we are to apply CMM/CMMI and establish the Teaching Capability Maturity Model (T-CMM).

However, as it can be expected, the development of T-CMM needs to address the applicability issues raised in this paper. Besides, in the future study, the T-CMM would be used in the real educational environment. Empirical issues such as the cost and effort for conducting teachers' T-CMM assessments, as well as the continual development of the model, ought to be addressed as well. It is hoped that through the discussion venue of this prestigious conference, valuable comments and suggestions can be obtained for helping the development of the model.

6 Acknowledgement

The paper expresses an appreciation to Mr. Kuo for his assistance and the earlier contribution to this research. The authors thank the Taiwan National Science Council for granting and sponsoring the research (98-2410-H-008-015). We also thank the anonymous reviewers for their contributive comments.

7 References

[1] ACM, AIS, AIPT (2002). *IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*, Association for Information Systems.

[2] Barnett, R. (1992). *Improving Higher Education*, SRHE and Open University Press: PA.

[3] Chen, C.Y., Chen, P.Y. and Chen, P.C. (2007). A CMMI-based, institutionalized method for continually managing information system quality improvement. *Journal of Quality*, 14(3), 267-283.

[4] Chen, C.Y. and Yu, P.L. (2008). An e-business CMMI model with case studies in the cosmetic and drug industry, *Journal of e-Business*, 10(1), 139-172.

[5] Cramer, K.M. and Alexitch, L.R. (2000). Student evaluations of college professors: identifying sources of bias. *Canadian Journal of Higher Education*, 30(2), 143-164.

[6] Daigle, R.J., Longenecker, Jr. H.E., Landry, J.P. and Pardue, J.H. (2003). Using the IS 2002 model curriculum for mapping an IS curriculum. *Proceedings of ISECON 2003*.

[7] Dounos, P.K. and Bohoris, G.A. (2007). Exploring the interconnection of known TQM process improvement initiatives in higher education with key CMMI Concepts, *the 10th QMOD Conference*.

[8] George, R.T. (1997). *Academic Freedom and Tenure*, Rowman and Litlefield Publishers, Inc.: Oxford.

[9] Grant, D., Mergen, E. and Stanley. W., (2004). A comparative analysis of quality management in US and international universities. *Total Quality Management*, 15(4), 423–438.

[10] Horine, J.E., Hailey, W.A. and Rubach, L. (1993). Transforming schools: total quality management in higher education. *Quality Progress*, 26(10), 31-38.

[11] IWA2 (2003). *Quality Management Systems - Guideline for the Application of ISO 9001: 2000 in Education*, International Workshop Agreement 2.

[12] Louden,W. (2000). Standards for standards: the development of Australian professional standards for teaching. *Australian Journal of Education*, 44(2), 118-134.

[13] Lutteroth, C., Luxton-Reilly, A., Dobbie G., and Hamer, J. (2007). A maturity model for computing education. *Proceedings of the ninth Australasian conference on Computing education*, 66, 107-114.

[14] Marsh, H.W., & Roche, L.A. (1997). Making students' evaluations of teaching effectiveness effective: The critical issues of validity, bias, and utility. *American Psychologist*, 52, 1187-1197.

[15] Martin, V.A., Hatzakis, T., Lycett, M .and Macredie, R. (2004). Building the business/IT relationship through knowledge management. *Information Technology Cases and Applications*, 16(2), 27-47.

[16] McKeachie, W.J. (1997). Student ratings: The validity of use. *American Psychologist*, 52, 1218-1225.

[17] McRoberts, H.A. and Sloan, B.C. (1998). Financial management capability model. *International Journal of Government Auditing*, 25(3), 8-11.

[18] Paulk, M.C.; Curtis, B.; Chrissis, M.B.; Weber, C.V. (1993). Capability maturity model, version 1.1. *IEEE Software*, 10(4), 18 -27.

[19] SEI (2010). *CMMI for Development v1.3*, Carnegie Mellon University Press, PA: Pittsburgh.

[20] SEI (2006). *The SCAMPI Method for CMMI evaluation*, Carnegie Mellon University Press, PA: Pittsburgh

[21] Seidin, P. (1997). *The Teaching Portfolio IN Higher Education*, Anker Publishing Company, Inc.: MA.

[22] Sen, A., Sinha, A.P., and Ramamurthy, K. (2006). Data warehousing process maturity: an exploratory study of factors influencing user perceptions. *IEEE Transactions on Engineering Management*, 53(3), 440-455.

[23] Singh, G. (2002). Educational consumers or educational partners: a critical theory analysis. *Critical Perspectives on Accounting*, 13(5), 681-700.

[24] White, B.A., Longenecker, H.E., Leidig, P.M., Yarbrough, D.M. (2003). Applicability of CMMI to the IS curriculum: a panel discussion, *Proceedings ISECON 2003*, 3515, 1-5.

[25] Williams, P. (2008). A practical application of CMM to medical security capability. *Information Management & Computer Security*, 16(1), 58-73.

Using Games To Teach Computer Programming

Frederic Allen IV, Chong-Wei Xu, and Victor Clincy
Kennesaw State University, Kennesaw, GA, United States

Abstract – *With the software industry growing, colleges and university are under pressure to graduate computer science professionals. However, many of these degree programs fail to instill proper programming skills. Students are taught how to put applications together without the knowledge of stable, professional software design practices. Even more bothersome, is the fact that many entry-level software engineers lack the basic foundations of to correctly program in an N-tiered architecture environment. The problem lies in the teaching methods used to graduate computer science students. All degree programs rely on traditional methods of having the student program boring desktop applications. To succeed, a student not only needs to be challenged, but also be given a sense of accomplishment. Game programming can provide students with a superior skill-set that will guarantee success in the software industry by demonstrating proper application design, and most importantly, by giving the student a challenging, enjoyable project.*

Keywords: Gaming, Teaching, Programming, Java

1 Introduction

Using games to teach students the intricacies of computer programming has become the latest trend. Even at their simplest level, games are complex systems of interlinked data structures; it is this complex nature that lends itself best to teach students proper program design, and implementation of advanced programming topics. Moreover, games make learning fun; students are more inclined to apply learned knowledge to creating a game than they are to creating traditional software applications [4].

2 Minotaur and the CoreFramework

In developing Minotaur, there were many aspects of software design that had to be considered to create a working game. In particular, these aspects covered proper design, and polymorphism; polymorphism by itself was difficult because it is not a software paradigm that is used on a daily basis. Basing the knowledge gained from classroom instruction to an interesting problem, this paradigm was much easier to understand; gaming helped in grasping a software paradigm that is usually taught in a boring, abstract manner. In addition to different software paradigms, gaming also taught that a specific discipline could be used to help tackle common software problems.

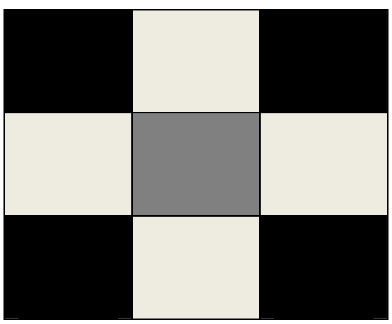
Computer games have very different class designs than typical desktop software; because of this unique design the game forces the student to solve a problem not encountered in the daily process of writing software. Minotaur uses a simple, but complex linking of classes to create a reusable class hierarchy; this modular approach is not too dissimilar to data access layers used in business logic libraries, and is a very important concept to learn to use correctly. Throughout the course of designing Minotaur, many difficulties were encountered that had to be overcome, and some special features had to be developed to make for a more robust game; however much research was done on the history of the game to find a starting place for this project. Many of the difficulties encountered in Minotaur focused on artificial intelligence, portable debugging, and good Object Oriented design; and it was these three features that drove the creation of Minotaur. Good Object Oriented software design is paramount in any project; however for games, it is of even more importance than usual. The reason behind this is that games encompass a very sophisticated system of driver classes, and underlying sub-systems used for logic and updating; the larger the game, the more complex these interconnected components become. Without a good framework, the cost of maintaining the game's code base becomes tiresome.

By their very nature computer games are highly coupled, a direct violation of good Object Oriented design; but a solid architecture will help mitigate the downsides of coupling by allowing the programmer, and those who must maintain the code, a portable "engine" that sits underneath game specific logic. Minotaur uses as its engine the CoreFramework, a modular framework of basic screen logic and input. The CoreFramework was used for all course projects preceding Minotaur, and it has worked flawlessly; very little modification was made during Minotaur's development because of the framework's modular architecture.

3 Artificial Intelligence

Minotaur's artificial intelligence was not as easy to implement as the CoreFramework was; the chief difficulty being trying to figure out how the enemy was to chase the player around the board. There exist several methods available to achieve this procedure, but the one chosen was a simplified version of what most games use. The most common method of path finding is called A* (pronounced A star) [3]. Using this method, Minotaur would pre-calculate the shortest path to the player, and follow that route; what was found was that because of the predictable nature of the movements options available to both the enemy and the player, A* produced a lot of computational overhead. The solution was to use a path finding algorithm called nearest neighbor; it is very similar to A*, but produces computations in real-time based on a very small sample set of movement options. In Minotaur's case, both the player and the Minotaur can only move in four cardinal directions; this makes it very easy, and very fast to compute which square would take the Minotaur closer to the player's position.

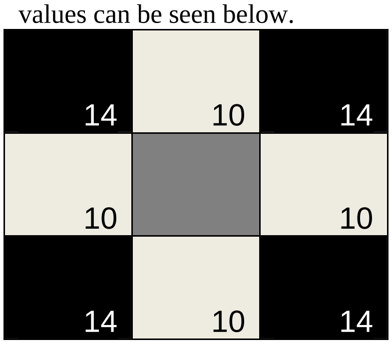
The nearest neighbor algorithm works by calculating the smallest movement cost of all the squares nearest to the Minotaur’s position, below the gray squares represent valid moves, while the black squares represent illegal moves.



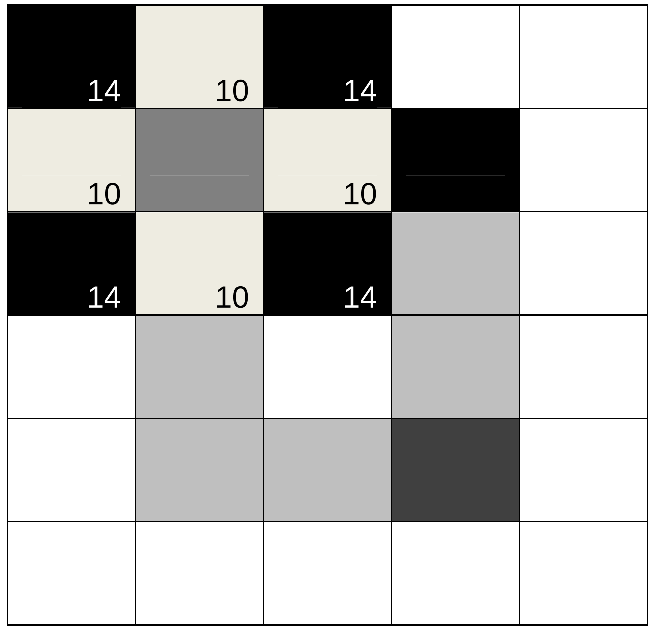
By restricting the Minotaur to only four possible moves, the path finding algorithm is greatly simplified. To calculate the F cost or movement cost of each valid square, Minotaur uses the following formula:

$$F = G + H.$$

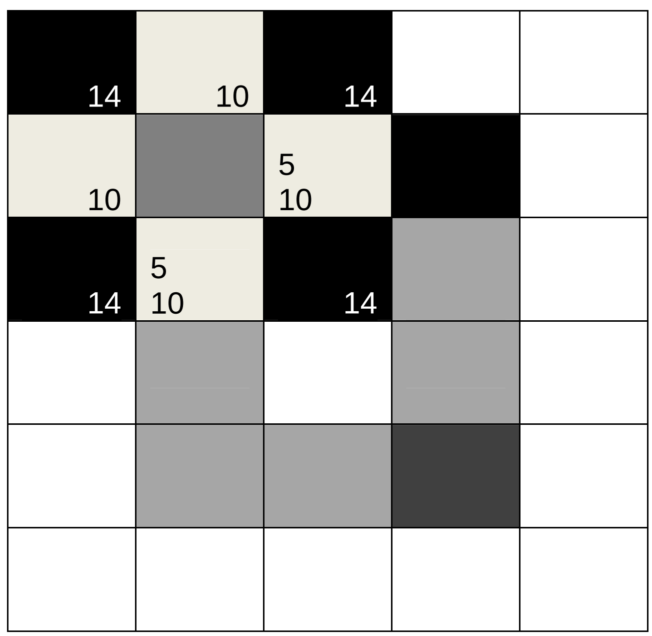
To obtain F, we need two additional formulas, one to calculate G, and the other to calculate H. G can be found by assigning a value of 10 to all adjacent squares, and a value of 14 to diagonal squares [2]; using the same graphic, the G values can be seen below.



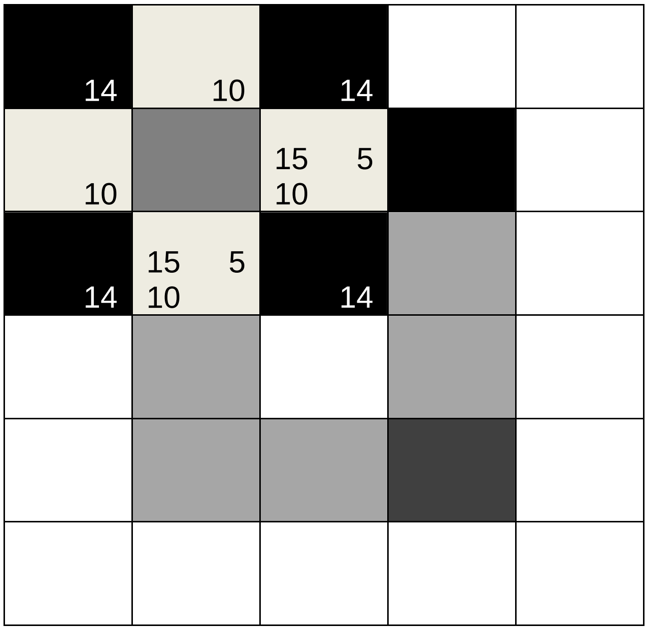
H is calculated by finding the number of squares between the Minotaur and the player, the graphic below depicts what the game board would look like with the player represented by the dark gray square, and the Minotaur by the lighter gray center square, black squares are obstacles, and gray squares are possible paths to the player.



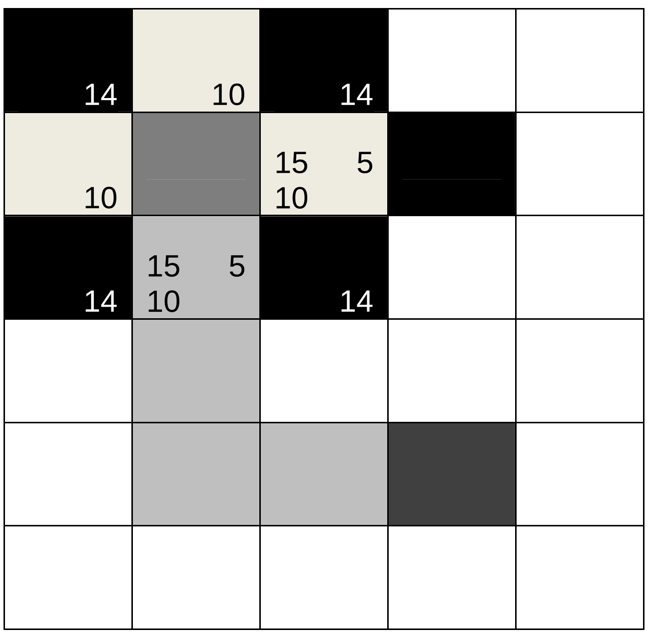
What we can see is that disregarding obstacles, the Minotaur is five squares from the player in both a south movement and an east movement; now we assign the H value of five to both of those path’s respective squares to get the graphics below.



Now we can calculate the F cost by adding G and H together for all squares; the resulting F costs are shown below.



By ignoring squares with obstacles, we get a final path to the player shown in gray below.



In the event of a tie, such as would be the case here had there not been an obstacle, Minotaur simple takes the last path calculated as the best route; for example, if the east movement had been the last F value calculated, Minotaur would have moved the enemy along that path instead of the alternative southerly path. What is truly great about using nearest neighbor is that all F cost values can be re-calculated in real-time. Because of this real-time calculation, Minotaur is able to adapt to player moves with a finer degree of precision than traditional path finding algorithms.

4 Program Logging

In addition to the easily portable framework, and the fast real-time path finding, Minotaur also has a rich program debugging mechanism. Throughout development, it was found that using the console for standard output was easy, but what was needed was a system that could live inside the game, and log errors to a file for viewing. The most human readable format for viewing files is HTML; the reason for this is that the markup allows the programmer to create a structured, stylized log where errors can be quickly identified by colored text. The final version of the program log feature allows quick output to a UTF-8 encoded HTML file that is XHTML 1.0 strict compliant for cross-browser compatibility.

Since Minotaur's log uses a single class for output, with an inner enumeration of constants for different text colors; the result is a highly modular, self contained debugging log that can be use in any project. Also included is automatic handling of HTML escape characters in the event markup needs to be written out to a file.

The most significant portion of ProgramLog.java’s code can be found in the Ouptut(String,eLogColor) function. This function takes two parameters, a string which represents output text and a constant from an inner enumeration that defines various colors. A random access file is used to rewind from the end of the file, insert the output string, and then append back the data that was rewound over. After the log file is opened for writing, the last 16 bytes of data which correspond to the </body> and </html> tags, along with two newline characters are stripped out, the output string is appended to the file, and the closing tags are written back to the end of the file. What this does is allows the programmer to log out to a file any text he wishes, and be able to view the log file in any web browser.

5 Graphics

Although Minotaur does contain features of good structure, and easy logging, there exists one area of its architecture that is complicated, and could not be easily debugged. All of Minotaur's drawing takes place at the GDI level, there are no fancy sprite batch classes or ManagedResourcePools; because of this, drawing is slow, and often times unpredictable. The most severe problem encountered was trying to clear the screen to a default clear color in between levels; in Java, this is actually more difficult than it sounds. In OpenGL there exists glClear() [5], DirectX has a GraphicsDeviceManager wrapper class with an associated clear() function; but Java has neither of these functions natively. In Java all drawing must be done in an overloaded paintComponent function; the wisdom of this is that all components know how to repaint themselves, so any drawing overtop of these components will be in their paint function.

The downside to this is that paint calls become bloated, and difficult to debug because of multiple threads acting on the same function call.

Even trying to separate custom drawing code into different functions in an inherited child class, erasing all image data from the screen was a chore; this can be attributed to Java's internal structure: mainly the running thread, and its associated actions. Below is Minotaur's main game thread, notice the use of a Boolean sentinel to control the thread body [1].

```
public void run(){
    running=true;

    try
    {
        while(running)
        {
            if(!isPaused)
            {
                updateGame();
                repaint();
                Thread.sleep(80);
                Thread.yield();
            }
        }
    }
    catch(InterruptedException ex)
    {
        System.out.println(ex);
    }
}
```

What is most bothersome about this structure is that the frame rate of the game is based on the thread sleep interval, and the repaint function can block the current thread execution if it takes too long. Nearly all professional games use a different approach: a single rendering function with update and repaint calls to object managers that redraw themselves onscreen. What is different about this method versus Java's implementation is that all components are updated and repainted by the application's base class with a constant framerate that syncs paint calls, not thread processes or timers.

Another key difference is that all professional games have the ability to force repaint calls at any time. Java does have the repaint() function that forces components to redraw themselves, but it is lacking in the fact that components can choose to ignore it if the event dispatcher thread is busy. Not having direct access to hardware level function calls is a serious drawback to the validity that Java is a viable gaming platform; that is not to say that Java games are poor in design, but they should be used for instructional purposes only, to get an understanding of proper application mechanics.

Professional games do not use Java GDI calls to implement rendering updates. Since Minotaur has its redrawing code in the paintComponent function, its architecture does suffer, but this should be overlooked in the classroom as a language design flaw, not an application design flaw. A better design would be a core drawing method called by the JVM not the game such as XNA rendering [6] code:

```
protected override void Draw(GameTime gameTime)
{
GraphicsDevice.Clear(Color.Black);

//game specific rendering code

base.Draw(gameTime);
}
```

Sadly, Java does not have anything remotely close to this within its native architecture; in order to achieve similar results, developers must rely on third party implementations such as JOGL or Oracle's own Java 3D, a framework that is not installed automatically by the Java SDK. However, both of these graphics frameworks are prone to excessive boiler-plate code and other implementation specific details; neither of which is good for teaching clean application design.

6 Input

One other area of Minotaur's architecture that appears as problematic is user input. When tested across four different machines: a Windows 7 desktop, a Windows XP desktop, an iMac desktop, and a MacBook Pro laptop, Minotaur performed flawlessly. Yet when demonstrated on a Windows XP desktop on campus, no keyboard input was registered by the framework. This case is frustrating as a developer because Java is supposed to be "write once run anywhere", yet there seems to be a few exceptions to this phrase. After analyzing the code, no errors could be found, and in fact, the code was tested the day prior to demonstration; same code, different machine, different results.

What can be most frustrating is that Java's object model should not allow this to happen. The Java programming language has a very rich interface mechanism where abstract event classes can be attached to components or class definitions. This bridging of events and event owners creates a solid framework for delegating responsibility to the correct owner without resorting to passing around objects or creating cumbersome message pump functions, a problem seen in the Win32 WndProc function.

Even so, Minotaur demonstrates that this bridge is not always as solid as the developer would believe; current .NET applications use a similar binding feature by exposing callback delegates for each component, Java abstracts this concept of callbacks by allowing a developer to register an event listener on just about anything. The problem lies in that if the component does not raise the events it is registered to handle, then the code will not be executed, there is no way short of reading the Java documentation to know if a component will fire a given event.

One more problem with this abstraction is that all input is ultimately handled by the target machine; that is, all Java events are eventually passed down to machine level calls. In order for Java input to work correctly, events must be handled in a proper fashion, namely by components having “focus.” What was found was that on UNIX systems, components do fire events even if they do not have current input focus. For example, running under OS X Snow Leopard, Minotaur does fire KeyEvent events even though the game panel does not explicitly call requestFocus or requestFocusInWindow. The same code on a Windows machine refuses to raise KeyEvents; the code fails because Minotaur did not call the requestFocus function [7].

This machine dependant inconsistency can make the Java language frustrating, and debugging user input for games a tasking experience; yet again this difficulty should be used as a learning experience where students see that not all code functions the same across multiple operating systems. From attaining this concept, students learn to develop for all possible outcomes, not just those seen on the development machine.

Despite a few problem areas, Minotaur can still be used as a teaching tool; basic principles of object oriented design such as polymorphism and object passing can be seen in Minotaur's design, and those principles can be translated into the classroom as raw concepts, not just standard implementation practices. Too many students are learning how to build complex programs without a real understanding of how the underlying concepts that make up the programs function. By using Minotaur as a teaching tool, students learn how to break applications into modules; they gain understanding on proper logic design, and most importantly, they have fun while learning. To better illustrate the point of Minotaur's design being used as a teaching aid, consider the game's most powerful feature, object passing.

7 Objects Abound

Minotaur's use of object passing is a fine example of how not understanding the basic concepts can have a detrimental effect on programming capabilities. Most of the modular programs written today focus on data access, and are more services than actual programs, so unless a student is only going to develop services, he needs to understand objects. Since legacy programs that use a modular design rely on object passing for logic updates, students must know how to construct and pass objects throughout various parts of a program. Examples of this design are the manager classes that comprise the bulk of Minotaur's design. The MapManager class handles all logic related to the game board, but it must also know where the players are at the same time. It gets this information by a public function that takes the player's current position as an object parameter, and stores the information for later use. Without this kind of object passing, Minotaur would have to rely on a call-back delegate whose implementation would be rather awkward.

Modern MVVM software does use call-back delegates, so this paradigm is important to learn as well, but a student must first start with an understanding of older technology before he can grasp the newer software design fundamentals. After analyzing how Minotaur handles its data passing and updates, student can use this knowledge in their own projects, and be better prepared to enter the job force knowing that the concepts they have studied will be applicable on a daily basis.

Another way that Minotaur can be used to help teach students is by its structure. Properly structured programs, be them games or otherwise, have clearly defined functions that perform a single task. Program logic should be easy to follow, and broken up into distinct units so that it is easy to maintain and debug. Minotaur has followed this principle by segregating CRUD operations, or create, read, update, and delete operations, into separate functions. Again, not only is this good programming practice in the sense that each function does a single task, but it makes the program easier to maintain, and code maintenance is a topic not currently taught in universities.

Program maintenance is something that very few students ever think about, and fewer professors teach, but it is crucial to daily software development. Rarely will one person work on a project, maintain the project, and then simply let it die. Software projects are usually implemented in teams with each person doing a portion of the development; they are also constantly being updated, often by people who had nothing to do with the original design. One important lesson that Minotaur's framework can teach students is that one must always plan as if someone else were going to take over maintenance of the code tomorrow; thinking in this manner will help students build a loosely coupled framework.

8 Conclusions

Daily software development depends on upon the ability of its designers to create efficient, modular frameworks with the least cost of maintenance; Minotaur can help students approach these development concepts in a manner that is fun, and intuitive. Too often the basic concepts are glossed over in favor of teaching students "how to program" instead, the real focus should be on “how to program well” for real-world scenarios. While not all scenarios encompass gaming, games can be used as an effective teaching tool to help students master important software development concepts before they enter the job force so they are not at a disadvantage.

3 References

1. Davison, Andrew. Killer Game Programming in Java. Sebastopol: O'Reilly Media, 2005. 11-33. Print.

2. Three Rings Design Inc. “Path Finding Tutorial”. December 23, 2006. http://wiki.gamegardens.com/Path_Finding_Tutorial

3. Lester, Patrick. “A* Pathfinding for Beginners”. July 18, 2005. http://www.policyalmanac.org/games/aStarTutorial.htm

4. Harlow, David. “Games as an Educational Tool”. April 6, 2004. http://www.gamedev.net/reference/articles/article2082.asp

5. OpenGL API Documentation http://www.opengl.org/documentation/

6. XNA API Documentation http://msdn.microsoft.com/en-us/library/bb203873.aspx

7. Java Documentation http://download.oracle.com/javase/1.5.0/docs/api/overview-summary.html

Educational Multiuser Online Game, "debugger" game for Introductory Computer Science Class

Ilmi Yoon[1], Gary Ng[2], Zoran Millic[1], Supakit Kiatrungrit[3], Yiyi Miao[4], Sunggye Hong[5]

[1]Computer Science Dept, San Francisco State University
[2]Pacific Ecoinformatics and Computational Ecology Lab
[3]Dept. of Information Technology, San Francisco State University
[4]OPTWAT. INC, San Francisco
[5]Dept of Special Education, San Francisco State University

Abstract

Computer games became daily routine to young generation. Especially games with social interactions are building strong cultures to young students. Multiplayer Online Role Playing Game (MMORPG), "debugger" is developed to utilize this new culture for education. Introduction to Computer Programming is a challenging course for majority of students as concepts are abstract, non-intuitive and completely new to most of them. Considering the serious decline of CS major students across US, developing publicly available educational games for CS students is a timely effort. Debugger game aims to build a collection of educational mini games within a community of learners. Within the virtual world of debugger game, players fight with bugs by solving problems, play "CodeGame" to earn resources to build their own motherboard space, interact with other players via chatting, friends list, reputation list and other features. The debugger game is intended to take advantage of social interactions and community to retain players longer, promote players to solve more quests, and encourage players to discuss and learn from each other more actively, as recommended in peering learning or peer tutoring. The debugger game is up and running and a game client can be downloaded for playing at http://thecity.sfsu.edu/~debugger.

1. Introduction

Game-based learning has been one of active educational research area with hopes that playing game would effectively aid students' learning. Large number of studies showed the educational impact of the game-based approach, educational games (or called as edutainment, serious game, gamification) have been adapted and used in diverse areas such as elementary math to professional medical trainings (Green '03 & Prinsky '01). Recent study shows that game-based education reaches beyond original objective of educational level and motivated players to achieve much more – promote the players potential upto extraordinary level (Cooper '10).

Cooper utilized multiplayer online game as social interactions serve well to motivate players. Other studies also showed significance of social interactions in the game to retain players longer. Figure 1 shows the general play time of stand-alone game vs. MMORPG game; MMORPG game keeps players longer over time (Ng '05). We observed that popular MMORPG games are extremely good at retaining their players for very long period time (months to years) and invite friends to play together (spread through human network). Recently, Farmville, a relatively simple farm nurturing (crops or farm animals) face book game made a nice show case that how quickly a game can spread through social interactions (50 million active players) and keep players to play over and over.

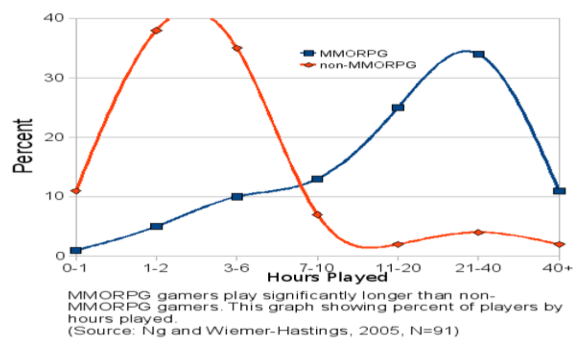


Figure. 1 - MMORPG vs. non-MMORPG games, hours played; Source: Addiction to the Internet and Online Gaming.

Popular MMORPG games like World of Warcraft has several millions players. Lots of research has been done to analyze the main driving factors (Song '07 & Yee '02). These studies pointed out that players especially enjoyed the fact that there is a community of audiences to give compliments when player achieved high level or completed quest. It is an important finding which can significantly increase educational games' efficacy. Educational games that are well designed for good educational impact often suffer from the fact that players do not play voluntarily over and over.

The MMORPG game, "deBugger" was designed and developed to make use of these findings to assist Introductory Computer Science course. It is generally accepted that it is very difficult to learn to programming (Jenkins '02). For example Bergin and Reilly (Bergin '05) note that *It is well known in the Computer Science Education (CSE) community that students have difficulty with programming courses and this can result in high drop-out and failure rates.* (p. 293). The concepts are non-intuitive or/and overwhelming. Students need lots of exercises to digest the concepts, close interactions, guidance and help when they are lost. Average pass rate for this course at colleges across US is around 60% (Bennedsen '07). Considering that 70% decline of incoming college freshman intending to major in CS, it is imperative to develop and provide publicly available educational game for CS students.

Debugger game is MMORPG game there is persistent virtual worlds that continues to exist even after a user exits the world, and players are represented by their characters in that world. The characters have permanent status like game level, virtual money, health, list of friends, game items like weapons or tools.

In the debugger game, there are different styles of games that players can play with other players or against bugs; debugger title was inspired by the origin of the word (removing a bug to fix errors). These mini games (games inside of debugger game) are intended to develop competence in computer science concepts when played repeatedly over and over.

Design principles of debugger game are; (1) Make use of players attachment to the character. In commercial MMORPG games, players have shown strong attachments to their online status (or their own characters). To be able to level up their characters, players tolerate grinning process –known as boring, time consuming and repetitive tasks. Players happily spend hours and hours to succeed in a quest. While there is pure pleasure of succeeding in quests, studies show that players are strongly motivated by the fact that their successes in quest result in level-up or acquisition of special awards. This motivation can be easily linkedd to Educational MMORPG game. (2) Studies have shown that students learn from peers as much as from teachers (Kaufman '99, Christudason '03, Johnson '93, Falchikov '03). Students tend to ask questions for clarifications among peers first and then ask questions to teacher if no answer can be found from peers. In addition, peer pressure pushes students not to be left behind. Online game communities have shown that experienced players take great pleasure helping novice players. These advanced players can be further motivated to help new players like a TA, giving close individual interactions that teachers may not give in real world. Frustration from the learning can be alleviated by sharing the similar troubles of individual with peers. All these peers can be connected in the virtual world, available when needed, and increasing educational impact. (3) For shy students, they can hide behind their virtual character, so do not need to fear the failure. (4) Importantly, usual MMORPG games grow and adjust depending on the needs or desire of the player community unlike console games that have pre-fixed everything before manufacturing. Debugger game can do the same. In addition, debugger can collect all sorts of (educationally) meaningful player activities in the game server. These data can be very useful for iteratively design cycle (design, development, user test) of the debugger game for various users to maximize their learning outcome.

One thing to note is that debugger game is not intended to replace the ordinary class teaching, but to be used as an effective aid like a virtual TA or lab, so students can comprehend the core pedagogical components and then progress smoothly. The core idea of debugger game is helping students to master the concepts through voluntary repetitive practices (lab) and clarifications from other players (virtual TA).

Section 2 discusses the mini games and community features within the debugger game and section 3 presents the structure of the debugger world. Section 4 discusses on-going implementations and user trial plan and section 5 concludes the paper.

2. Mini games for Computer Science Learning

There are different approaches to introduce the computing concept to beginners. First and newer approach is starting with abstract and high level, usually utilizing objects like graphics, sounds, and scripts to get quick results and motivating to try further. Examples are MIT Scratch, CMU (storytelling) Alice and NYU Rapunsel. These are tangible (video+audio) and fun approach to problem solving and logical flow. However these approaches are still experimental and eventually CS students have to take programming course. So, still majority of college and AP CS courses take traditional approach of using practical programming languages like JAVA or C++. And CS AP Collegeboard uses this approach. When the second and traditional approach uses practical programming languages, courses cover the core concepts (problem solving with pseudo-code, variables, data types and operators, selection flow control, repetition flow control, methods, strings, arrays) in sequential order with slight variations. Mini games are designed to work for one concept at one time or one group of a few related concepts only, so each game can be designed with specific focus.

First mini game called GodeGame is for practicing program structure, importance of syntax rules, and simple program flow. When mini game starts, player will receive a simple mission randomly assigned according to the player's level and a list of possible answer segments will be displayed on the left panel as shown figure 2. To solve the mission, player picks up the answer segments that are correct in syntax and in proper program structure then arrange them on right answer panel in proper sequence by drag and drop. With this game, player does not have to solve the problem from scratch, but think of the functionality of given answer segments and then place them in right order. The answer segments can be either actual program code level or abstract high level like pseudo code as shown at figure 2. Abstract high level helps students learn problem solving using top-down or divide-conquer approach easily without thinking of the syntax level. Screen capture of playing this mini game is recorded and available at YouTube (YouTube link).

Figure. 2 – CodeGame Interface: mission is displayed at upper left corner and the options to drag are displayed below the mission. Individual options can be dragged and dropped.

Another mini game currently implemented is multiple choice questions as it is easier to implemented and used for user trial quickly. Multiple choice style is easy to ask diverse topics from definition, concepts and picking a correct execution result from a given code segments. These questions are organized into levels in the order of variables, data types and operators, selection flow control, repetition flow control, methods, strings, and arrays. So students can play the right level that they have learned.

To make the multiple choice questions to be fun, these questions are used while player is fighting with bugs in the battle, playing board game with other players, doing a quest with big boss bug with other players or use the question for PvP (Player vs. Player) game. Rewards for playing these games are increasing health, getting game gold, increasing level or getting a gift of game items. Game items can be also purchased with game gold. And game items are designed to add fun to the game and allow players to grey out two options from the multiple choices, protect player's health while being attacked by bugs or upgrade decorative items.

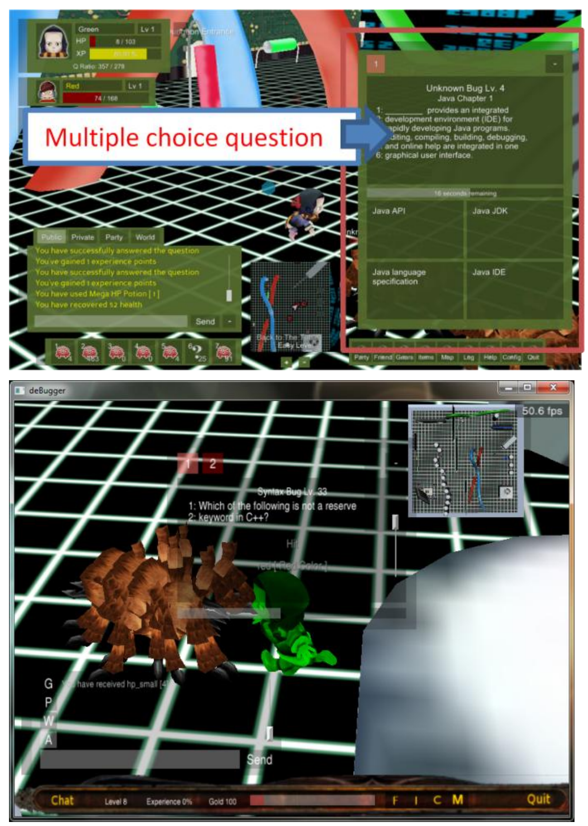


Figure 3a & 3b – A player is fighting with a bug. Bug asks a question and the player has to answer. Health gets damaged as time lapses. Player's health, level, gold are shown at upper left corner. Lower left corner panel serves as a message panel and chat panel.

Figure. 4 – Two players are playing a board game. Each player takes turn, roll the dice, and solve a problem to progress. Winner who arrives the end point faster receives game gold as reward.

In addition to those game features, there are many social activity features within the debugger game. Players can chat with other players in a few different ways. Public chat is for chatting with anyone in the same virtual space (room) and private chat is for chatting with ones that a player wants to chat. Also chatting with friends who are online is possible. A player can manage friend's list. Player can choose to show their level and other performance to the friends or public.

Figure 5 – Player can check their inventory for game items. Players can purchase game items using game gold, trade with other players or give a gift to friends.

Player's achievement can be measured by levels, number of questions played and the accuracy ratio. Achievement board displays top 10 high score achievers to encourage players to play longer and better (Figure 6).

3. DeBugger Game Architecture

Social gaming and MMORPG are relatively recent game genres as they bloom over the availability of Internet. Social interactions within the game fire the excitement of the game and popular games easily develop communities of multimillion players. Once community of learners is created, students can learn not only from educational game components but also from their peers discussing their problems with them. Debugger game was created with the vision of nurturing such community of learners playing collection of games together, help each other and invite more friends to join.

Figure. 6 – Player's performance is measured in diverse ways. And players can see other players' performance unless blocked by the player.

To enable this, we need to create and support virtual world persistently utilizing game server. Game server runs all the time, connects all the clients, and updates DataBase Server as well as clients. Game client connects to the game server when a client wants to play the game, receive the latest update of the virtual world since the last time player logged out, and interact with players already online. Database server stores all the data of individual players (health, level, money, game items, friend's list, performance, and etc) and other data to maintain the game world persistent. Game protocol is developed to make the communication effective. DeBugger game also has a bug server that maintains all the bugs – how often they appear (spawn), what kind of game item they drop, how aggressively attack players, etc. Bug server also controls bugs with simple AI to make them wander around and handle collision naturally to make the game fun when player interacts with bugs.
Figure. 7 shows a flow of debugger game, showing each component and their connections to each other. Game client was developed utilizing Panda3D and python scripts. Game server was developed using JAVA, using MySQL as DB server. Bug server was extended from game client that already included collision handling.

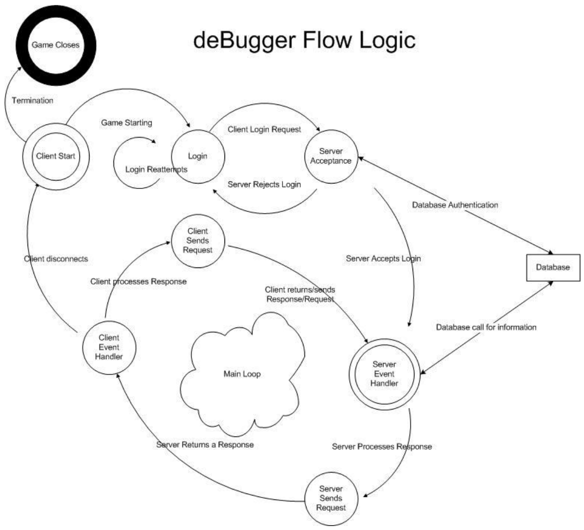


Figure 7 - deBugger Flow Logic used to show the cycle of information being passed between *Client* and *Server*

Currently, debugger supports two big virtual spaces (a desk and a motherboard) where players can meet and interact with each other and bugs. Figure 3b shows mother board space where players can explore and fight with bugs. Like other MMORPG games, this virtual space can be extended over time by collecting feedbacks from players.

4. On Going Implementation

Currently additional mini games are designed and being implemented and user trials are under IRB review for human subject protocol. We are trying the iterative game development cycle of design, implement and user test followed by a new cycle of revised design, implementation and user trial. The first user trial will focus on the CodeGame, bug battles, and other community features discussed in section 3.

Designs of additional games are as follows. Four more mini games have been designed. First mini game to be implemented is called “Variables” and designed for practicing concepts of variable, data type and operators. Students need to learn that variables are spaces in the memory where program stores value according to the data type and operators act on the values according to data type. This will be similar like Cisco Binary Game [ref] where user finds the memory space and fill the value according to the type. For example, 1 / 2 is 0.5 according to usual math, but 1 / 2 is zero in JAVA, C, C++ while 1.0/2 is 0.5. Repetitive playing of this game will help students master the three concepts combined together. Second game is for practicing selection flow control concept. It will utilize idea of racing game and player’s route will be selected depending on the value in the car and the condition at the branch. To be able to finish the lap in given time, player should choose the value wisely to utilize the conditions at the branch to stay at desired routes. This game will be extended to repetition flow control by running multiple laps. For the beginners, the car move very slowly to give enough time to think and the car will get accelerated as player progresses well. Third game is for practicing repetition flow control, especially nested repetition flow control using pattern printing style arcade game. Forth game is for practicing array and repetition flow control.

Strength of MMORPG approach for educational game is that game server can collect every activity of player; how long, how often, how many questions does player attempt to solve, what is the accuracy rate, and then how their performance at class correlate to the game performance.

5. Conclusion

Introduction to Computer Programming is a challenging course for majority of students as concepts are abstract, non-intuitive and completely new to most of them. Considering the serious decline of CS major students across US, developing publicly available educational games for CS students is a significant contribution. Debugger game aims to build a collection of mini games within a community of learners. The game will be iteratively refined through user trial and revision of the game to increase the educational efficacy. Debugger game tried to maximize the education efficacy utilizing the social interaction and the power of the community itself (players help each other). Once the human subject is approved, user test will be performed on the debugger game.

ACKNOWLEDGEMENT

This project is being funded by National Science Foundation Div. Of Biological Infrastructure, Biological Databases and Information, NSF DBI-0543614

6. References

(Bergin ’05) S. Bergin and R. Reilly. The influence of motivation and comfort-level on learning to program. In Proceedings of the 17th Annual Workshop of the Psychology of Programming Interest Group pages 293-304, University of Sussex, Brighton UK 2005.

(Bennedsen ’07) Jens Bennedsen and Michael E. Caspersen, Failure rates in introductory programming, SIGCSE 2007, pp 32-36.

(Christudason ‘03) Christudason, A. Successful Peer Learning, http://www.cdtl.nus.edu.sg/success/sl37.htm

(Cooper ’10) Cooper, at al, Predicting protein structures with a multiplayer online game, Nature 466, pp 756-760, August, 2010.

(Falchikov ’03) Falchikov, N., Blythman, M., Learning together: peer tutoring in higher education.

(Green ‘03) GREEN, C S, AND BAVELIER, D. 2003. Action video game modifies visual selective attention. Nature 423 (2003), 534-537, Letters to Nature.

(Johnson ’93) Johnson, D.W.; Johnson, R.T.; & Holubec, E.J. (1993). Circles of Learning. Edina, MI: Interaction Book Company.

(Jenkins ’02) T. Jenkins. On the Difficulty of Learning tp Program. In Proceedings for the 3rd Annual conference of the LTSN Centre for Information and Computer Sciences , Loughborough, UK August 27 - 29, 2002.

(Kaufman ‘99) Kaufman, D.B.; Felder, R.M.; & Fuller, H. ‘Peer Ratings in Cooperative Learning Teams’. Proceedings of the 1999 Annual ASEE Meeting, ASEE, Session 1430’.

(Ng ’05) Ng, Brian D. and Peter Wiemer-Hastings, Addiction to the Internet and Online Gaming, Cyber Psychology & Behavior 8.2 (2005)

(Papstergiou ’09) Papstergiou, M., Digital Game-Based Learning in High School Computer Science Education: Impact on Educational Effectiveness and Student Motivation, Computers & Education, v52 n1 p1-12 Jan 2009.

(Prensky ’01) PRENSKY, M. 2001. Digital Game-Based Learning. McGraw-Hill, New York. ISBN 0-07-136344-0.

(Song ‘07) Seungkeun Song, Joohyeon Lee (2007), Key factors of heuristic evaluation for game design: Towards massively multi-player online role-playing game, International Journal of Human-Computer Studies, no. 65, 709-723

(Yee ’02) Yee, N. (2002) Understanding MMORPG Addiction

(YouTube link) http://www.youtube.com/results?search_query=SFSU+debugger&aq=f

Developing a MMORPG game in One Semester

Ilmi Yoon[1], Gary Ng[2]
[1]Computer Science Dept, San Francisco State University
[2]Pacific Ecoinformatics and Computational Ecology Lab

Abstract

Building a game has been used as effective way of teaching computer science. Building a Massively Multiplayer Online Role Playing Game (MMORPG) makes another step and serves as a great comprehensive tool to teach important aspects of technology, teamwork and software engineering principles. A course was taught to design and develop a working MMORPG within one semester. The whole class was structured in to several teams (Game Concept Design Team, Client Team, Server Team, Database Team, Art Support Team, Game Contents Team, Testing Team, Launching Team and IT Support Team) and students needed to join one or two teams. Each team presented their progress, discussed future milestones and troubleshoots, updated documents for clearer communication and utilized SVN throughout the semester. Unlike usual class setting, all students worked collaboratively together like one company to achieve the goal. In one semester, students started from concept design and completed a working MMORPG called "deBugger" (http://thecity.sfsu.edu/~debugger), learning game design, 3D graphics, Game Engine, Server-client architecture, Game Protocol, network programming, database, Software Engineering principles, and large application development as a team project.

Keywords: MMORPG, Game development, Industry style team work, team project taught class

1. Introduction

Game design and development has become a popular computer science course over recent two decades [1]. As students' enthusiasm to build game has been so strong, instructors utilized the eagerness to teach many aspects of computing, including computer graphics, artificial intelligence, human-computer interaction, security, simulation, and software engineering [2]. Parberry [1] studied what game companies commonly want, so their game course to be designed to teach the experience; (1) work on a large project, (2) creation of game demo, (3) team player, and (4) learn independently and taught them in game class. These previous work shows the promising potential that students are willing to take a tough and intensive course work to achieve the goal of building game.

Social gaming and MMORPG are relatively recent game genres as they bloom over the availability of Internet. Social interactions within the game radically increase the excitement of the game further and popular games easily develop communities of multimillion players [3]. Therefore, building a MMORPG game can motivate CS students to focus on the course work and increase learning outcome substantially. Unlike stand-alone game, a MMORPG game keeps virtual world persistent even after players log out, so game server has to run infinitely connecting all the clients and updating game status at database. Therefore, building a MMORPG game covers broad spectrum of computer science technology from computer graphics, 3D modeling, game engine, network programming, client-server architecture, and database, so it helps students exposed to diverse technical components comprehensively. Also a MMORPG game is indeed a large application and building one ensures students to learn the qualities that software companies want.

In section 2, we discuss the course design to teach students to build a working MMORPG game within a semester (15 weeks). In section 3, we present the results and learning outcome. Conclusion and future work are discussed at section 4.

2. Course Design

This class was designed for SFSU CS senior and graduate students who completed programming languages, data structures and Software Development Principles but not necessarily have taken elective courses such Software Engineering, Network nor Database courses. Considering that majority of students find jobs at industry after graduation, the class was organized in more of industrial flavor; a team of students (1) receive tasks and milestones, (2) achieve the milestones by actively looking for solutions from any available resources, (3) interact with other teams collaboratively, (4) and produce documents for clear communications between teams and future extensions.

The class was organized into Game Concept Design team, Game Server team, Game Client team, Game protocol team, Database team, Art Support team, Game Content team, Testing team, and Launching team. Within first 2 weeks, objectives and the responsibility of each team were discussed and then students were assigned into one or two teams based on their interest and backgrounds. From then on, teams worked on in parallel to achieve their milestones, while learning all the required technology for the given task and sharing their learning with other teams.

Summary of each team's task is as below.

Game Concept Design Team: this team has most important initial responsibility. The objective of the whole class was to build an educational MMORPG for Computer Science students, especially for beginners who are taking first programming courses. The concept design team should consider the limitations such as time (10 weeks for actual implementation), lack of artists, and unknown factors like the potential of other team's performance. The team also has to collect everyone's idea, finalize the game concept into several milestones and produce the documentation.

Game Client Team: this team is responsible to implement 3D game client using Panda3D. Students have to learn 3D graphics, Panda3D game engine and Python. Also client communicates with server using the game protocol to update the status (position, motions, and levels, etc) of other clients concurrently in the same virtual space.

Game Server/Protocol Team: this team is responsible to implement game server that connects to all the clients, synchronize every client status and update the Database. Team did not start from scratch as there was a JAVA game server written for a similar MMORPG game called "NurseTown" by students of Internet Application course. The server team had to read and revise the game server to work for "deBugger" game. Also the server team has to develop an effective set of protocols used between clients and server. Students in this team have to learn socket programming, client-server architecture, and Network protocols.

Game Database Team: this team is responsible to design the DB schema, install mySQL server and deploy the DB for every activity to maintain virtual persistent world. Students in this team have to learn SQL, Database scheme design and mySQL server management.

Art Support Team: this team is responsible to provide 3D models to create the virtual world and characters in the game. As there are no artists in the class, this team has to find freely available 3D models and convert the format to work with Panda3D. Students in this team have to learn basic of 3D modeling tools such as Blender or Maya and converting 3D model formats, and real time rendering performance trade-off (visual quality vs. rendering speed).

Game Content Team: "debugger" game is a unique MMORPG game for educational purpose. Therefore, the educational game contents are as important as other game assets. This team develops educational contents. As senior or graduate students of computer science, this team is already familiar with educational contents that beginner programmers learn which is good and bad. This team needs to interview beginner students, observe the troubles that they probably forgot and produce effective educational contents.

Game Testing Team: this team is responsible to create scenarios for testing the stability of client, server and protocols. One main task is to develop an autonomous client that connects to the server and simulates an ordinary client. Then the scaling of such clients can be used to test the server and protocols. The students in this team have to learn Panda3D, network protocols and practices in software testing planning.

Game Launching Team: this team is responsible to develop a web site where client installer can be easily downloaded and players can get simple instructions about how to get started with games – game rules, help, hot keys, etc. Also this team works on packaging the client program into an easy installer program.

IT Support Team: this team is responsible for helping every team to function without any system/software issues. For example, this team sets up SVN and creates account for each student. This team also helps teams to set up development (IDE) environment.

There were a little over 20 students in the class, so each student choose one primary team and one secondary team, so each team has at least 2 to 3 students. By having students in a few different teams, students could learn more than one technical area and teams can communicate and update the progress easily. Each team has to work very collaboratively as the whole class works for one project together.

Important concepts and technologies were taught to the whole class by several lectures, but most of class meetings were used as presentation of milestones, progress, trouble shootings of problems on hand, so the whole class stays in a same page and gets exposed to the problem and solutions.

Software Engineering principles have been practiced throughout the whole course as teams realized the importance of the inter-team communications due to the rapid developments in parallel. Changes in one team did propagate changes in milestones or requirements in other teams. SVN was actively used.

3. Results

The whole class successfully completed a working MMORPG game. Each team achieved milestones close to the initial (very ambitious) goal. **Game concept team** came up with the title, "deBugger" and the theme of the game; fighting against bugs within inside of computer, inspired by the origin of the word and the significance of the word to computer science students. Team also prioritized concepts in to several milestones, so kept the milestones doable for one semester, but left many interesting ideas for future extensions.

Fig. 1 – Theme of the game.

Primary milestone for Fall 2009 was to build virtual world where players can explore and fight with bugs by solving multiple choice questions. Health drops under bug's attack and game items can be used to shield from the bug's attack (delay the health damage or clear out options from the multiple selections). Players can level up by solving questions required by each level. Also player should be able to chat, create friends' list and maintain their inventory (trading or giving gifts).

Art Support Team achieved beyond the requirement. Students learned 3D modeling tools and created 3D environments for the game. We used 3D character models from Panda3D repository, but all the environments (figure 2, 3 & 4) were created by art support team and theme was inspired by movie Matrix and Tron during inspirational discussions at class.

Fig. 2 – Full view of the desk space. This is a default/initial space where players get started.

Fig. 3 – A corner of motherboard. This is a dungeon where lots of bugs live. Players can teleport into this space from the initial desk space.

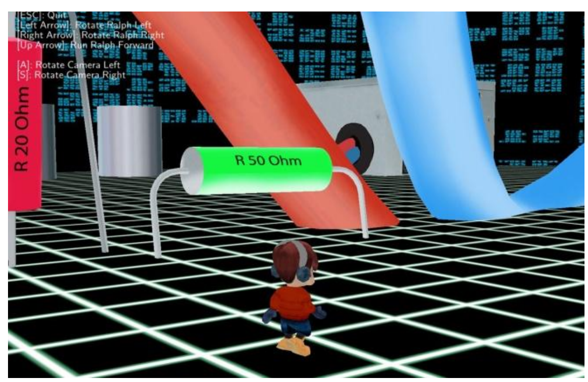


Fig. 4 – Another corner of motherboard.

Client Team was the largest team and they were divided into sub teams to be responsible for specific tasks in parallel. List of such tasks are as below.

i. Registration Process
    a. Avatar Selection
ii. Login Process
iii. Chat
iv. Bubbles (translucent panel)
    a. Chat Bubbles
    b. Character Name and Damage Bubbles
v. Friends
vi. Inventory (figure 5)
vii. Character Info
viii. Hot keys
ix. Camera control
x. Character Movements
    a. Interface (mouse, Keyboard)
    b. Collision
xi. Battle System (figure 6)
xii. NPCs (Non Player Character) (figure 7)
xiii. Bug
xiv. Mini maps
xv. Mouse Picker

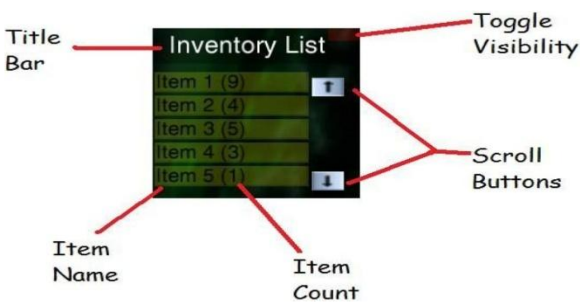


Fig. 5 – Inventory of a player.

Fig. 6 – A player is fighting with a bug inside of computer by solving the quiz. On the right upper corner, there is a mini-map showing the current location of the player. At the bottom, there is a bar where user can click to activate several game functions such as chat box or user inventory.

Fig. 7 – NPC (Non Player Character) requires simple AI to interact with players; guide players to follow rules of game or inform about events to participate.

Fig. 8 – Bugs are attacking players. Bugs have AI to chase the player, handle collision and re-spawn.

Fig. 9 – Health bar shows the damage from the bug's attack. This player lost all the health.

Client team also achieved impressive progress of completing all the requirements and the resulting game client was reliable and capable of very smooth rendering. Each task listed above took intensive effort to understand the technology (3D graphics) to manipulate Panda3D game engine.

Server team worked hard to make MMORPG game alive. Server team was able to reuse the majority of server code built for a game called, "Nursetown" by Prof. Ilmi Yoon and her graduate students. Server team had to understand the communication mechanism (figure 10 & 11) and modified DB schema and added lots of new game protocols.

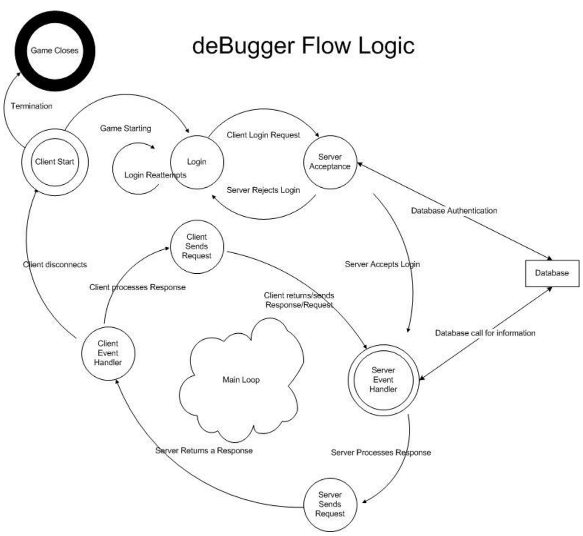


Fig. 10 - deBugger Flow Logic used to show the cycle of information being passed between *Client* and *Server*

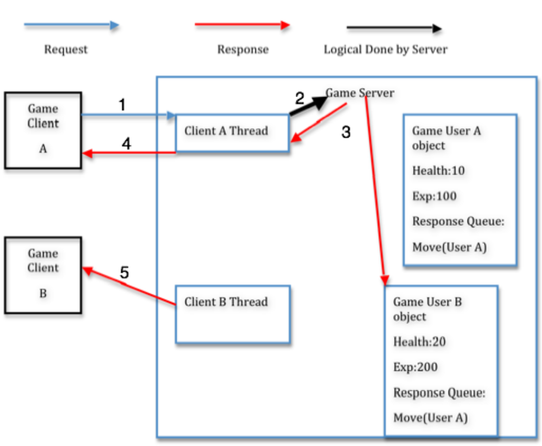


Fig. 11 - Handling of Request and Response from Game Client to Game Server. In the picture, starting at 1 the game client A sends the game server a request. The game server receives the request from its game client object. At 2 the game server handles the request and creates the corresponding response. At 3, a response is sent back to the original requestor (client A), and also same response is created for client B to be updated about client A. At 4 the move response is sent back to the player. At 5 the client B sends a heartbeat (10 times per sec) request and receives the update of client A's move.

Game Contents team focused on adding educational contents into the game. They attended "Introduction to Computer Programming" class to refresh the memory of what they learnt at the beginning level and interviewed students of the course. Game contents team structured pedagogy of computer programming learning and created quizzes in different levels for such purpose.

Data Base team studied design of efficient DB schema and developed it for debugger game. Team also installed mySQL server and deploy the data sets.

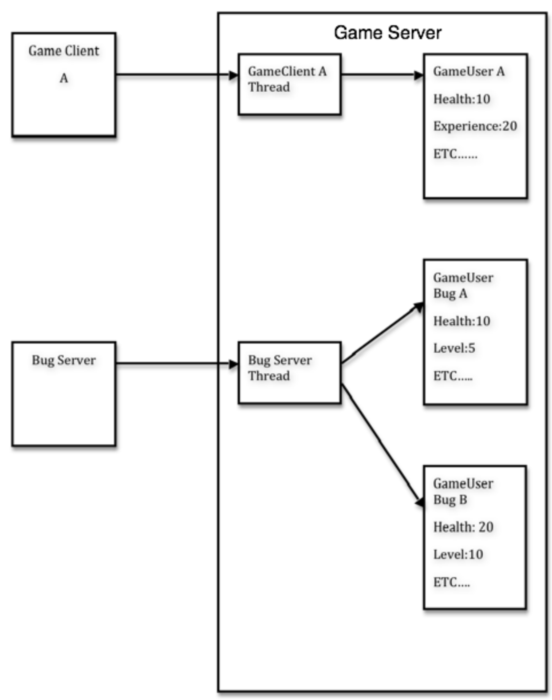


Fig. 12 - Difference between Player and Bug Server overview

Testing team was developing a test client that has a minimal AI to wander around, acts like a usual client and save the communication log, so we can use the test client to measure the scalability of the server. In the middle of the semester, we found that the debugger game needs a bug server that controls all the bugs that has AI to attack the player and wander around the debugger world. Testing team took the role and successfully completed the creation of the bug server with the close collaboration with the client team (figure 12).

Launching team studied popular MMORPG hosting company's web site and designed a web site for the possible full version of the debugger web site and implemented core set of it to be able to launch the game for programming class students (figure 13). The launching team made a simple web site (http://thecity.sfsu.edu/~debugger) with link to download client application (self-extracting installer for windows), game rules, information of how to get started, and screen captures.

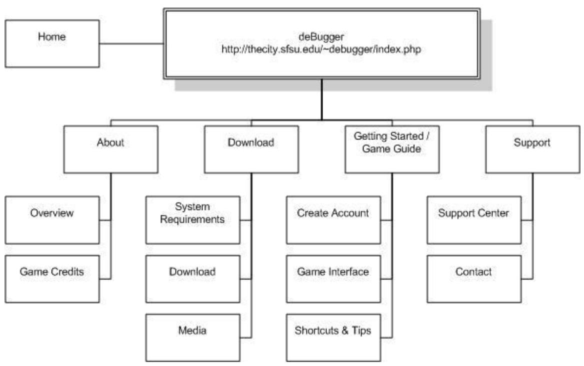


Fig. 13 – debugger web site map : debugger launching team studies other MMORPG site and designed a core functionality in similar style.

The class as one whole team completed a working game (version 0.9) by Dec. 2009. The game contained core MMORPG game features. The client and server connect and communicate in stable manner. DB stores persistent data. User can create an account with options of a few characters. And then user can log in at the location where they logged out last time. Player can explore the virtual world starting from a desktop space into the inside computer world. User can find if friends are logged in and chat with friends or other players within the same space. Players can check their game items at the inventory box, health bar and the current level as well as their current position at the mini-map. Most importantly, players can battle with bugs using the multiple choice questions and the health bar and level were properly updated.
At the same time, whole class produced one giant document (437 pages, available at http://thecity.sfsu.edu/~debugger/download_documentation.php) of each team's work for future developers who will extend this game to the next level.

4. Conclusion

Students were proud of what they achieved in one semester (15 weeks). Considering that the first few weeks were used to the introduction to the class and setting up the team, students had about 2.5 months to work on the game. To be able to make the MMORPG game work, each team has to deliver what they were expected in milestones. It serves as a good pressure on each team and students collaborated across team enthusiastically. Most students put efforts beyond requirements, resulting in good learning outcome.

This class can serve as a good team project course for senior students in Computer Science, so they can taste the industry style problem solving approach and team work in academic environment. Also as a by-product, the class produced a usable game that is currently used in CSc 210 Introduction to Computer Programming course at SFSU. The objective of using this game for educating CS freshman seemed to give pressure to students to deliver a smoothly working game as their clients are sitting right next to them, so it lets them alert about their end user.

To be able to replicate the same class in the future, course material needs to be better collected and organized. There are hundreds of discussion threads at iLearn news group that were effective while being used, but not so organized, so may not be transferrable to next year class nicely. Inter-team communications were well captured to be used in the class for each team progress presentation and documentations. But inner team communication and resources were not saved. Each team created very useful resources to be shared within team to learn the required technology for each team for example, client team learned Panda3D, Python, collision handling together helping each other. Same for each team. Vast amount of self study materials were created by each team. These can be better managed to be created at wiki, so they can produce more re-usable study resources.

Acknowledgement

Big thanks goes to Zoran Millic and Supakit who developed Nursetown game and provided kind support to CSc 631/831 students to understand the key technical components. Sergey Sergeev as an art support team leader went beyond the requirement and provided creative 3D models to the game. Jason

D'Silva and Gary Ng put tremendous hours to lead the client team to produce the client with all the required features. Alan Nguyen worked hard to make the bug server possible. The rest of the client team that includes Alexander Bonilla, Julio Benitez, Nicholas Byrd, Vivekanand Rao and Yi Liu contributed great effort to implement several important features used in the client. The database team that includes Alvin Wan, Calvin Kuang and Tun Win worked great together to design and implement a database used specifically for the server. Ali Ugur, Jordan Mangini and Sara Tily produced an interesting game concept that we all could follow. And Kaven Tan for contributing the best he could to get the server running.

This project is being funded by National Science Foundation Div. Of Biological Infrastructure, Biological Databases and Information, NSF DBI-0543614

5. References

[1]. R. M. Jones. Design and implementation of computer games: A capstone course for undergraduate computer science education. In Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education, pages 260–264. ACM Press, 2000.

[2] I. Parberry. Introduction to Computer Game Programming with DirectX 8.0. Wordware Publishing, 2001.

[3] R. Moser. A fantasy adventure game as a learning environment: Why learning to program is so difficult and what can be done about it. In Proceedings of the 2nd Conference on Integrating Technology into Computer Science Education, pages 114–116. ACM Press, 1997.

Teaching Algorithms:

Backgammon Resolution Using MiniMax Strategy

L. Felice and M. Ciolfi Felice

Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil, Argentina

Abstract - *A System Engineering curriculum should involve elements or concepts that reflect a trend towards distinguishing the true professional software from the occasional programmer. In order to introduce some fundamental concepts in an early stage of a System Engineering career, we propose to teach game programming. It could be considered as a sophisticated software development in an undergraduate curriculum because it involves a wide range of knowledge and skills. This article proposes the reasons for teaching game programming and show the application of a game resolution as a global project in a second year of the career. Also, the methodology used throughout the course, which introduces good design principles, is detailed. In particular, a solution using a MiniMax strategy to the Backgammon game is presented.*

Keywords: Algorithm design techniques, MiniMax algorithm, Backgammon game, programming teaching.

1 Introduction

A System Engineering curriculum should involve elements or concepts that reflect a trend towards distinguishing the true professional software from the occasional programmer. This trend has important consequences for universities. What matters is to teach students fundamental ways of thought that will accompany them throughout their careers and help them grow in this ever-changing field. As Bertrand Meyer analyses in [1], a software curriculum should involve five complementary elements:

- principles: lasting concepts that underlie the whole field;
- practices: problem-solving techniques that good professionals apply consciously and regularly;
- applications: areas of expertise in which the principles and practices find their best expression;
- tools: state-of-art products that facilitate the application of these principles and practices; and
- mathematics: the formal basis that makes it possible to understand everything else.

We use a methodology based on the points mentioned above to teach two courses of “Analysis and Design of Algorithms” -ADA1 and ADA2- in the second year of the System Engineering career at Universidad Nacional del Centro de la Provincia de Buenos Aires ([2]; [3]). The methodology is used in both courses and starts at abstraction level defining the problem domain and identifying the involved abstract data types (ADTs). They are organized in libraries to enable their reuse. Abstraction mechanisms are of paramount importance in object-oriented programming [4], which is a programming paradigm to define the abstract data types and their relationships.

At implementation level, the data types and the algorithms that use them are implemented in C++ language ([4]; [5]). These algorithms are based on problem-solving techniques that are taught throughout the first course such as Greedy method, Divide and Conquer, Backtracking, and Dynamic Programming. The second course deals with Graph Theory, Searching problems, NP-C problems, and the basic strategies and algorithms to their resolution, always working with the complexity concepts and linked with abstract data types.

The problem solution to be defined depends on the student’s knowledge about programming languages, algorithms design techniques and available development environments (IDEs). The learning is mainly based on problem solving. On the one hand, the students solve practical exercises corresponding to each topic of the course.

On the other hand, they develop a capstone project based on game programming, which allows students to integrate the learned concepts and thus to prepare them to work in a professional environment. This project is developed as final examination of the courses.

This paper is organized as follows: Section 2 summarizes the methodology used in the courses. Section 3 briefly introduces the motivation. Section 4 presents the Backgammon resolution. Section 5 draws the project organization and, finally conclusions are made in section 6.

2 The methodology

The goal of ADA1 and ADA2 is to introduce the basement for software development, basically: ADT specification, algorithms design techniques for P problems, Graph theory, NP problems, Geometric and String Matching algorithms, and the construction of component libraries for later reuse. In order to achieve this goal, the proposed methodology starts at abstraction level, which describes a problem independently of the data type representations and of a particular programming language, and concludes at implementation level with efficient programs written in C++ language.

At abstraction level, the problem domain is defined and the entities that intervene in the problem are identified. The classes of objects are identified and implemented in C++ language. The organization in libraries of the classes facilitates the later reuse.

At implementation level, the elements of the problem are translated to C++ language code and integrated with algorithms that intervene in the solution of the problem. Throughout ADA1 the students solve practical exercises that use the type library and apply different techniques of algorithm design such as Greedy method, Divide and Conquer, Backtracking and Dynamic Programming. Basically, during ADA2 course they work with Graph algorithms, Aproximation and Heuristics algorithms solving practical exercises and laboratory practices. Also, they learn Geometric and String Matching algorithms.

The curriculums of the courses are a result of an ongoing effort at our career to teach computer science to young students, addressing a broad selection of topics while maintaining depth and rigor. Students learn several programming languages that correspond to different paradigms. In the first year of the career, the principles of structured programming are taught and Pascal is the programming language used. In the third year, the course of Object Oriented Programming introduces the basic and advanced concepts of the paradigm using Java language. For this reason, in both of the courses, C++ language is used to establish a connection between structured and object oriented programming. C++ is a hybrid object oriented language that allows working with object classes that belong to the problem domain and functions that manipulate them in an independent way.

As a result, many students have enough knowledge about C++ language through the integrated programming laboratories of parallel academic courses and subjects.

The strategies used to apply the methodology and to get succesful results are based on a constructivist learning. It is a process which involves complex interaction between previous knowledge of the student, social context and the problem to solve [6]. One of the strategies to reach is the development of a global project. This project, developed in groups of two students, integrates the main issues of the courses. This strategy encourages interaction among the students who work together to reach a common goal.

The global project is a large-scale development where the proposed methodology is applied to solve a game allowing students to investigate in the:

- identification of the ADTs that intervene in the problem,
- definition of the ADTs reusing type libraries and design,
- selection and combination of algorithm techniques to solve the problem, in particular, a game resolution, and
- programming and implementation.

In particular, in this article we show a project which deals with the resolution of Backgammon game based on MiniMax algorithm.

3 Motivation: Game resolution for the capstone project

Concluding ADAs courses, the first issue with which students are confronted when writing programs is “the problem”. Many times the evaluation of student’s examinations shows there are many of them who lack the ability to apply the concepts and techniques learned to unfamiliar problems. So, the confrontation with real-life problems to solve is a true challenge for them.

What programming topics might satisfy the proposed methodology and student motivation?

One of the effective ways of raising motivation and attraction of learning process is using games. Games are welcomed by students because a game generates an active and visual input. The player can continuously communicate with the game and the visualization and animation of the game can stimulate the player’s interest with a variety of dynamic scenes and objects. However, designing and implementing a game could be another story due to the fact that a game is a mixture of sciences, multimedia, arts, artificial intelligence, and so on [7].

Creating programs that can play games such as chess, checkers, and backgammon, at a high level is a challenge for students of a second year course considering that the vast majority of our students have taken an introductory programming course in the first academic year, typically in Pascal. Thus, our recent efforts were rewarded with increased student participation. The projects based on game development are a good way to introduce the core concepts of the courses. Game programming deals with many computations and has a lot of different objects interacting and moving.

Ending the second course, in the second semester of 2009, a project to develop the Backgammon game was asigned to a student. The objective was to apply the MiniMax search algorithm. To be accomplished, the student had done a little investigation and evaluation of the different proposed solutions for this game.

4 The project: Backgammon resolution using MiniMax algorithm

For games with two or more participants, the algorithm must choose a method for modelling opponent behavior. For example, in games with two completely competing players taking alternating moves, the model of choice is MiniMax, in which the opponent player is assumed to always choose the action that minimizes the possible loss. An illustration of MiniMax search executed on a two-player game tree is shown in Figure 1.

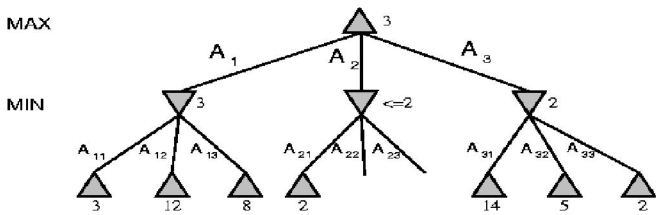


Figure 1: MimiMax search algorithm

The first succesful software for Backgammon game was BKG, by Hans Berline, which beated in 1979 the Backgammon world champion.

In the 80's Gerry Tesauro created Neurogammon, based in neural networks and supervised learning. This kind of learning consists of giving the program some domain information (codified by an expert) as an input, along with the expected output.

This approach was employed by TD-Gammon [8], which combined search, function approximation and self-play to create a high performance Backgammon agent, with minimal requirements for external domain knowledge. As a training method he used temporal difference learning, applied to the information collected after making the software play repeatedly against itself. The strength of this solution is not the depth of the search, but its evaluation function.

Another remarkable application is GNU Backgammon, a free software solution capable of doing strong game analysis and aiming players' mistakes [9].

4.1 The student's solution

The proposed solution consists of a MiniMax search with the inclusion of the Alpha-Beta prune and chance nodes to handle probabilities. The Alpha-Beta prunes the branches of the tree that won't affect the root valuation. When updating a node, if this value is beyond a threshold, it can be transmitted to its father as if every one of its children would have been analysed. It's convenient to sort the nodes so they can be pruned down as soon as possible, after exploring the first node of each level.

The presence of dice, on the other hand, makes impossible to apply native MiniMax, so it's necessary to include chance nodes between MAX and MIN levels. Hence, after every play, the 21 combinations of the two dice should be simulated, and then the excepted value for each one calculated. This would significantly increase the response time. The proposed solution is to randomly analyse a minimum of 14 combinations and then allowing to prune the branch. This number seems to be reasonable since it represents 2/3 of the set. Figure 2 shows the partial performance of the chance nodes applied to a variable with values between 0 and 1.

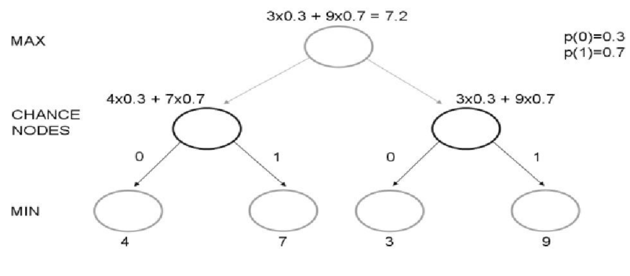


Figure 2: Performance of chance nodes

The valuation of terminal nodes (those that can be found at the depth given to the search) is determined by a *heuristic evaluation function* that combines common and advanced Backgammon strategies. The main feature of this function is its temporal context sensitivity and it will be described in the next section.

The solving algorithm is run with different breadths and depths according to the stage the game is in. At the beginning, there is a huge ammount of possible moves with low impact, so a deep backtracking search wouldn't retrieve significantly better results, and it would certainly take longer. In consequence, after a period of testing, the breath given to this stage is 8, and the depth is two.

As the game progresses, the possible moves become more decisive. At this point, the algorithm is run with a depth of 3 levels, analysing only the two more promising moves.

Since the same board state can be analysed more than once, it's very useful for the algorithm to keep a look-up table with previous calculated MiniMax values, in order to avoid doing the evaluation again.

Another improvement included in the proposed solution is to have stored the opening moves instead of calculating them. The reason behind this decision is that expert players in all over the world agree that for every first roll there's a more convenient move to make.

It's important to mention that the implemented heuristics aren't applied to the moves independently, but to a play (this is, the collection of moves a player makes in satisfying the requirements of a roll) as a whole entity, considering not only the state the board is after performing it, but also the temporal context of the game.

4.1.1 Heuristic evaluation function

Given a play and a player, the heuristic evaluation function returns the priority associated to that play. In order to do this, the moves that compose it are analysed together as a set.

The analysis is done according to their type (reentry, bear off, basic), since there are some specific heuristics that can be applied only to certain types of moves. This becomes evident in comparative advantages related heuristics that take the previous or the new situation as the comparison parameter.

For example, one of the designed heuristics consists of giving more weight to the play every time that one of its moves brings closer two checkers that were more than 6 points far from each other. Clearly, this reasoning can't be applied when reentring a checker, considering that it wasn't on the board before the move so it wouldn't make sense to calculate the distance from the other ones. This evaluation wouldn't be suitable either when analysing a bear off, because the checker would be off the board after performing the move.

As well as there are factors that raise the estimated value of a play, there are others that decrease it. It's important to indicate that every heuristic weight is set so that the final value reveals accurately the priority of making the play. Therefore, the evaluation function is oriented to create balance in situations where a very good movement is followed by one that makes the scene worse in the same proportion, and to show a tendency towards a play in which positive factors are higher than negative ones.

For instance, a play involving several bear off moves will have a very high priority, only got over by a play that covers a set of points on the inner board, helping to create a large size prime. The reason behind this decision is that it's strategically more convenient to block the opponent and prevent him from leaving his outer board, than to bear a checker off.

On the other hand, when a hit implies leaving a blot (or an empty point), the weight given to the move will depend on the point valuation. Thus, a very important one becoming vulnerable will negatively affect the evaluation of the play, and vice versa.

4.1.2 Examples

As mentioned before, the heuristic evaluation function is temporal context sensitive, since it considers the stage the board is in, and applies the corresponding heuristics. The most representative examples of this feature are:

- 7 point valuation: It's crucial to own it at the beginning, but as the game progresses, it loses its strategical power. Hence its valuation has to decrease when the game is close to the bearing in stage.
- Pure race stage: The players are no longer in contact, so hits are impossible and the goal is to get as soon as possible to the own inner board and bear the checkers off. At this point, most of the factors analysed on the contact stage don't apply anymore, and the priority of the plays will only depend on how suitable they are to make the player go ahead in the race.

The designed heuristics give more weight to a play if:

- It bears a checker off.
- It brings a checker into the own inner board.
- It takes a checker out from the opponent's inner board.
- It brings closer two checkers that were more than 6 points far from each other.
- It covers a point (with this priority order: 5, 7 - if it's not bearing in stage-, 4, a point on the inner board when

there are opponent's checkers on the bar, a point on the inner board, any other point).

- By covering a point it's helping to make a prime grow.
- It hits a checker (especially on the opponent's inner board).

Moreover, the heuristics penalise a play if:

- It separates two checkers that were less than 6 points far from each other (so one of them won't be reachable from the other with a single move).
- It moves a checker into a point that already had 5 of them (it attempts against proper distribution).
- It leaves a blot.
- By leaving a blot, it's decreasing a prime.

4.1.3 The implementation

The application was developed using TDAs and object orientation paradigm. It's been written in C++ language using Microsoft Visual 2005 ® as IDE.

Through its simple graphical interface it is possible to play against a human opponent or against the computer. Figure 3 shows a reduced class diagram for the solution. The full and detailed report about this implementation can be found in [10].

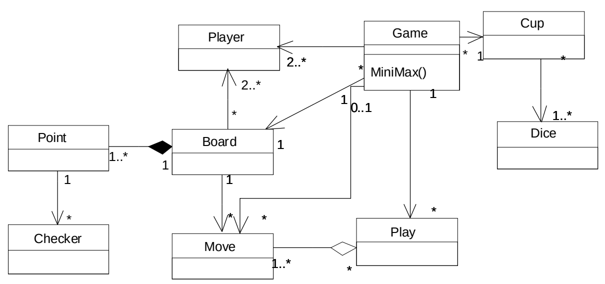


Figure 3: Class diagram for the solution

The design has a main class called Game, which has a Board (composed of a collection of Points), two Players and a Cup containing Dice. In each turn Plays are made by the Players, comprising a set of Moves that involve Checkers being transfered across the Points. The class Game is crucial in the proposed solution because it is where the MiniMax algorithm is run when the computer plays, and it's the only link between the logical and the graphical parts of the application.

5 Project organization: Practical aspects

The number of students registered in each course is approximately 130. The percentage of students that attend the final exam and those who have passes the partial examination, is aproximately 80% in each course. Taking into account the student profile, we proposed as final examination of each courses a capstone project.

Before choosign a topic (game) for the capstone project, a mentor group is assigned. The mentor, who belongs to the teaching staff of the courses, advises and helps the groups throughout the project development. The choosing of the project is related with the student interest and relevance to the teachers. The group-mentor contact takes place during the computer laboratory practices and a serie of interviews.

The duration of the development is about four or five weeks. The student must make two deliverables. The first containing the investigation made about the topic and the solution proposed, and the second containing the implementation itself. It consists of a detailed report and a CD/DVD with the source code and executable file. The mentor checks the deliverables of each stage and gives students a feedback. In case there were mistakes, students should resubmit the deliverables accurately.

Finally, each group presents the deliverable final version of the project to their peers and mentor. Each group member must demonstrate their involvement in the project to get through it.

6 Conclusions

Since we have been applying the methodology for both courses, we have evaluated the selection of projects to identify useful features across several games including Checkers, Go, Chinese Checkers, Monopoly, Puzzles, Video Games, etc. For each project, we follow the construction process made by the students and we collect students' opinions using the results of a set of projects made at the second year.

The experiment of teaching algorithms in this way has helped the teachers staff reach some interesting conclusions in relation to the effectiveness of the approach. Since 2004 the teachers staff of ADA1 and ADA2 courses have been working in this kind of capstone projects. Some of them were published at student's and educational symposiums ([11], [12], [13], [14], [15], [16]).

It's true that the work is hard but throughout these years, student's feedback has revealed that both classes of laboratory and mentor guide are essential to carry out successfully the project development.

Here, we have described the solution of Backgammon game as part of a capstone project for the course of Analysis and Design of Algorithm. Backgammon is a game involving strategy and chance and has received much attention for its solution. So, we consider that is not a trivial work considering the student preparation at the second year in the career. In particular, the resolution shown here is a good result and the game has a very good performance.

Game programming is interesting but tough and we think the development of a global project based with games allows students to integrate the learned concepts and to create a challenge to show good results.

7 References

[1] Bertrand Meyer. "Software Engineering in the Academy" ; Computer (IEEE), Vol. No. 34, 28-35, May 2001.

[2] ADA1. URL: www.exa.unicen.edu.ar/catedras/aydalgor

[3] ADA2. URL: www.exa.unicen.edu.ar/catedras/aydalgo2

[4] Bjarne Stroustrup. "The C++ Programming Language (Special Edition)". Addison Wesley, 2000.

[5] Nell Dale. "Programming in C++". Jones and Bartlett Publishers, 1998.

[6] David Jonassen. "Designing Constructivist Learning Environments"; M. Reigeluth, Instructional-design Theories and Models: A new paradigm of instructional theory (Lawrence Erlbaum Associates), Vol. No. 2, 215-239, 1999.

[7] Chong-wei Xu. "Why and How to Teach Game Programming"; Proceedings of the 2006 International Conference on Frontiers in Education: Computer Science & Computer Engineering (FECS 2006), June 2006.

[8] Gerry Tesauro. "Td-gammon, a self-teaching backgammon program, achieves masterlevel play"; Neural Computation, 215–219, 1994.

[9] GNU Backgammon site. URL: http://www.gnubg.org/

[10] Marianela Ciolfi Felice. "Resolución del juego Backgammon - Proyecto final"; Internal report for "Analysis and Design of Algorithms II", 2010.

[11] Iván Ridao Freitas, Santiago Vidal. "Algoritmos de resolución para el cubo de Rubik"; EST 2005 (34 JAIIO, Jornadas Argentinas de Informática e Investigación Operativa), 2005.

[12] Martín Fernández. "Algoritmos de búsqueda heurística en tiempo real. Aplicación a la navegación en los juegos de video"; EST 2005 (34 JAIIO, Jornadas Argentinas de Informática e Investigación Operativa), 2005.

[13] Fabricio Defelippe. "Algoritmos heurísticos de resolución para puzzles de piezas deslizantes"; EST 2009 (38 JAIIO, Jornadas Argentinas de Informática e Investigación Operativa"), 2009.

[14] C. Martínez. "Heurísticas aplicadas a la resolución del problema del Viajante múltiple"; EST 2009 (38 JAIIO, Jornadas Argentinas de Informática e Investigación Operativa"), 2009.

[15] Enzo Ferrante. "Clausula: Herramienta Didáctica para la Enseñanza de Lógica de Predicados de Primer Orden"; EST 2009 (38 JAIIO, Jornadas Argentinas de Informática e Investigación Operativa"), 2009.

Teaching Professional Ethics Issues Using Massive Multi Player Virtual Worlds

Amir Zeid[1], **Rehab El-Bahey**
azeid@auk.edu.kw
[1]Division of Science and Engineering, American University of Kuwait, Kuwait

Abstract – *Using massive multi-player virtual worlds for educational purposes is becoming more popular in the past few years. Second Life^{TM} is a 3D virtual world where users can create their own avatars and then use them to socialize, trade, travel or even go to universities in this virtual world.*

In this paper, we investigate using Second Life as an e-learning tool for teaching professional ethics courses. Through the real life simulation provided by second life, issues of copyrights, freedom of speech, privacy, and computer crimes can be experienced and tested in a virtual environment. This will give the course a practical dimension and will allow students to examine how computer ethics work in more realistic situations.

Keywords: Computer Science Education, using Virtual worlds in Education.

1 Introduction

Ethics is defined as the principles or standards of human conduct [1]. Computer ethics deal with how to make moral decisions while using technology whether in the workplace or in society in general. Professional ethics for computer science students is a major course in most of the computer science programs.

Wikipedia defines A massively multiplayer online game (also called MMO and MMOG) as a multiplayer video game which is capable of supporting hundreds or thousands of players simultaneously. By necessity, they are played on the Internet, and feature at least one persistent world. Massive multiplayer virtual worlds are becoming very popular. They are currently being used in entertainment, education, shopping and many other domains. Our idea is to map the concepts covered in the computer ethics course to virtual worlds to see how they can be applied, tested and analyzed. Virtual classes and virtual classrooms are becoming more common, as an increasing number of schools will hold online classes within Second Life. Courses that use second life as a simulation environment include economics, art, and architecture among others. In this paper we investigate using second life in professional ethics course for computer science and information systems students.

The rest of this paper is organized as follows:

- Section 2 introduces the environment where the proposed approach was applied. The research methodology is also discussed in this section.
- Section 3 introduces the findings and observations.
- Section 4 contains an assessment and discussion of our approach.
- Section 5 concludes the paper.

2 Background

2.1 Environment

We applied our approach throughout the professional ethics course offered by the computer science and information systems programs at the American University of Kuwait (AUK). AUK was established as a Kuwaiti university following the liberal arts philosophy. This necessarily makes it bi-cultural. So while the educational system is American, it exists in a very different cultural environment and that provides both challenges and opportunities.

2.2 Course Description

This course covers the social impacts, implications, and effects of computers on society and the responsibilities of computer professionals in directing the emerging technology. Legal, ethical, privacy, and security issues in computer usage are discussed. The course includes group work, student presentations, discussions, and student essays addressing the above issues. Case studies relating to computer ethics will be discussed. The course is a 100 level course. Most of the students are sophomores. The number of students in the class is around 25.

The following topics are discussed in the course:

- History of computing, computer applications and their impact
- Privacy
- Encryption
- Freedom of Speech
- Copyrights issues
- Computer Crime
- Computers at the Work Environment

2.3 Intended Learning Outcomes

On completion of this course the student should:

- Appreciate the meaning of professionalism and of professional ethics,
- Understand the expectations and responsibilities placed on professionals,
- Appreciate freedom of speech and understand censorship issues over the Internet content,
- Learn how to write and present a technical paper,
- Understand how technology could be used to protect privacy and intellectual property.

2.4 Technologies used

Second Life (SL) is a virtual world that has been developed and launched by Linden Lab on June 23, 2003 and is accessible via the Internet. A free client program called the Second Life Viewer enables its users, called Residents, to interact with each other through avatars. Residents can explore, meet other residents, socialize, participate in individual and group activities, and create and trade virtual property and services with one another, or travel throughout the world (which residents refer to as "the grid"). Second Life is for people aged 18 and over, while Teen Second Life is for people aged 13 to 17 [13]. Residents are allowed to build their own virtual objects. Linden scripting language could be used to produce sequence of events and complex actions. Residents are allowed to buy and sell stuff, get married, have their own business and most of real life activities. The Second Life Terms of Service ensure that users retain copyright for any content they create, and the server and client provide simple digital rights management functions. We also use other technologies in the course like Moodle [11].

2.5 Second life in Education

Second life could be used in education in many different ways. It could be used for basic presentations while the avatars attend virtual classes[8]. In addition, a three-way white board could be used to provide interaction during virtual classes [8]. Office hours of professors could be conducted in second life. Group meetings in collaborative software development could be arranged as well in second life [10]. Other simulation techniques including conferences, art exhibits, guest speakers and learning via games could be also incorporated depending on the learning goals. Educators should always make sure that second life should be integrated into the courses in a way to maintain the appropriate balance between the core content and the experiential learning [8].

2.6 Methodology

Students enrolled in the Computer Ethics course are asked to create avatars in second life. They can choose whatever name or orientation they want. Then they investigate the course topics and come back with concluding remarks about their findings. The topics are discussed in class as the course progresses. At the end of the semester, the students have to present their findings in a 15-minute presentation. They also submit a short report about their findings.

The experiment was carried out across 4 classes for the duration of 2 years.

3 Observations and Findings

Ethical issues including, privacy, freedom of speech, copyrights and computer crimes, were investigated for both the avatar and the creator of the avatar.

The following observations were noted down by students as each of the ethical issues was examined in Second Life:

3.1 Privacy

The following are some of the privacy issues that were investigated in second life. The findings show that there are many areas that need improvement to provide reasonable privacy for the residents of second life.

3.1.1 Privacy of avatars

Residents can simply know the date of account creation by looking at the profiles of avatars. To help protect the privacy of the avatars, some tools were created by individuals and companies. For example, avatars can buy private changing rooms. The irony is that some people created tools to invade the privacy of other avatars. Examples include scripts that enable avatars to listen to private conversations of other avatars.

3.1.2 Privacy of account holders

IP collection

Some people argue that IP addresses should be personal and private. It is implied in the privacy terms of Second life that Linden Lab collects IP addresses that the users use to connect to the service [9]. To get around this policy, some users use proxies to hide their original IPs.

Retention of Data

We tried to find out how long the data is retained after the cancelation of accounts (the data about the avatar and the personal data of users). We could not find clear record of that. In addition, Linden Lab states that should they ever file for bankruptcy or merge with another company, they may sell the personal information of its users to a third party or share personal information with the company with whom it merges [9].

Surveillance

The concept of invisible information gathering (Collecting personal information without person's knowledge) is discussed in the class. Mapping the concept to Second life we found out that Linden labs can see what the avatars are doing all the time. It is exactly similar to real life surveillance. There is no clear indication if this information is reused or not. Any avatar can take snapshots of the screen view without the consent of other avatars.

3.2 Fair Opportunity and Freedom of Expression

Censorship

The "Big Six" is a set of rules that control violations in second life. Incidents related to any of the six violations can be reported to second life administrators who can terminate the accounts of violating users. The six violations are following violations [13]:

1. **Intolerance:** This rule states that hate speech is not allowed in Second life. Also, the user should not engage in the following showing bad images, demeaning language, referencing others by race, ethnicity, gender, religion, or sexual orientation.
2. **Harassment** is forbidden.
3. **Assault:** The terms and conditions of second life state that second life is a safe environment. So, creating scripts that signify beating, pushing or shoving, or attacking other users in any form will be considered as assault.
4. **Disclosure:** Users are not allowed to disclose information about other users.
5. **Adult Regions, Groups, and Listings:** "Second Life" has been created mainly for adult users. But, in the main land (general open access land) it is prohibited to use any "Adult Content." So, there are designated places for "Adult Content" which have access control and age restrictions and verification.
6. **Disturbing the Peace:** Everyone in "Second Life" has the right to live in peace. Users that engage in repetitive advertising, sounds, or self-spawning that will slow the server performance or affect other residents will be penalized for peace disturbance.

Hate speech

If hate speech incidents take place, a violation could be reported. An example of hate speech events is an incident that was witnessed after an Algerian-Egyptian soccer game during the world cup 2010 qualifications. After the game there were claims that Algerians spectators harassed Egyptian spectators which resulted in a real diplomatic problem between the two countries [5]. What happened in Second life was similar. Some lands had signs making fun and using racist references to Egyptians by Algerians. Figure 1 depicts a sample of hate speech between Egyptians and Algerians in second life. In addition, white hate groups were reported in second life [12]. Neo Nazis also were reported to have activities in second life.
Observations concluded that second life provides reasonable protection against hate speech violations.

Figure 1 Example of Hate Speech in Second Life.

Protests

Demonstrating in real life in some parts of the world could lead to detention. Protestors could be detained in some parts of the world even if the protests are peaceful. Protests in virtual worlds could be a good alternative in such cases. We found out some protests in Second life against political policies. Examples include protests against Israel and the US policies in the Middle East. Figure 2 shows an example of protests in Second life. Palestinian Holocaust Memorial Museum in Second life is a very popular land where a lot of political activities take place.

Most of the observations confirmed that second life provides a great environment for protests.

Figure 2 Protests in Second Life.

Child pornography

Age verification allows adults to visit some adult-only lands. The process involves filling a form and an agreement that SL can use as a legal document in case of violations. The agreement states that: "The information that I am providing is true and correct, and I consent to its verification against public records or government-issued identification." Despite of this, violations of age-restrictions have been reported. In one of the incidents, Second Life was investigated by German police following allegations that some members are trading child abuse images in the online world [6]. To deal with such issues, SL introduced new age-classified lands like the teen second life for young teens from 13 to 18.

Global impact (gambling)

Gambling is illegal in some countries. For example, in Kuwait gambling is prohibited by law. All of gambling web-sites are filtered and blocked by the government. Virtual worlds like Second life can provide a backdoor to practice illegal activities in some countries. After receiving criticism from many countries, Gambling was totally banned by Second Life in 2007 [2].

Appearance, names and clothes

In Second Life, users have control over the appearance of their avatars. Users can pick names as well for the avatars. However, it was noticed that there are no Middle Eastern names in the available list of names. It was also noticed that there was no free clothes that support Muslim women dress code when second life started. This made Middle Eastern students feel that they were under-represented and deprived of equal opportunity rights in SL. Over our period of examination, students have started reporting that this issue was properly dealt with as SL started to introduced free women veils and other related fashion items. Figure 3 shows an example of free Islamic prayer clothes for women in the market place of second life [13].

Figure 3 Islamic Apparel in Second Life.

3.3 Copyright issues and Computer Crime

There are many ways to commit computer crimes in second life. Hackers can take the identities of other users, which is against the rules. Others will try and hack user accounts and use them, delete them or even sell them. This is against second life regulations because selling a user's avatar is just like selling one's identity.

Intellectual property

Second life follows the procedures described in the Digital Millinium copyright Act (DMCA) regarding copyrighted material [9]. There are many sensitive issues like designing games, lands, material artwork and many other creative works. Creative Commons (CC) has its own land in second life. The Creative Commons Second Life Salon's goal is to build a community in Second Life around Creative Commons licenses and standards through events and projects in the virtual world, Second Life.

Identity Theft and Copybots

Copybots are programs that allow users to make copies of objects without the permission of their creators. They have some limitations, yet using them can lead to a lot of violation of copyright issues. Using copybots is a breach of the Second life terms and conditions and could lead to getting banned from Second life [3].

The problem with copybots is that a large part of content-theft activity is accomplished with "hacked" or copybot viewers, an issue which is causing major concerns and dissatisfaction among SL users. Figure 4 shows examples of copybots in second life.

There are a lot of people who would try to hack other users. Also, there might be people in second life who create scripts that will expose other users or just hack them using these scripts. During the period of study, students found many tools to stop messages from copybots. There are methods to "mute" messages from certain suspected bots.

Figure 4 Copybots in Second life.

Hacktivism and E-Jihad
Hacktivism is "the nonviolent use of illegal or legally ambiguous digital tools in pursuit of political ends". E-Jihad. is one form hactivism. E-Jihad is the electronic version of the holy war (A religious war led with an exceptionally high grade of religious feeling) [7].

The major concern behind this practice is that Jihadists are using "Second Life" to recruit and also mimic real life terrorists [4]. Recently, ads started to appear for Jihad bomber vests that can kill all avatars within a certain distance from the avatar bomber. Having these types of tools available at acceptable and accessible media could alter the negative mind-set associated with them, which could be a major trigger of violence in real world. Figure 5 depicts a Jihad body bomber in second life.

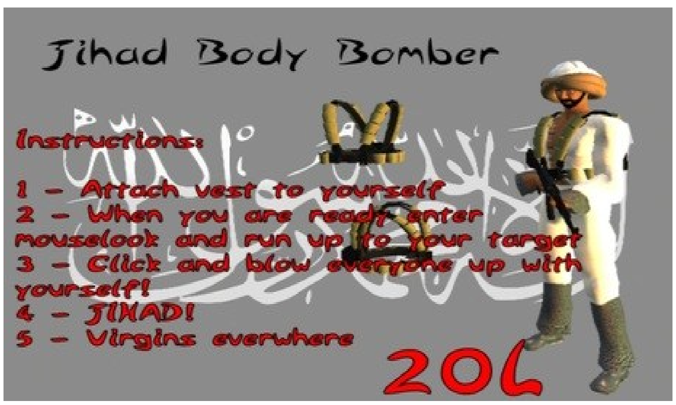


Figure 5 Jihad Body Bomber vest in Second life.

4 Discussion and Assessments

The outcome of our research clearly shows that virtual worlds; although still miss a lot of real life regulations and laws, can be an excellent tool for creating a more dynamic learning experience.

Using second life as an e-learning tool in ethics courses supports the learning goals of the course by enhancing the comprehension of professional ethics concepts and emphasizing the role of each individual in maintaining the ethical system in general.

In the conventional method of teaching the courses, we used to give students case studies to help them visualize how each concept work. But this method did not provide the students with the extra mile of being actively involved in changing unethical practices.

Second Life provides an interactive little world of ethical and non-ethical practices. It is not only a place to watch or observe action, but also a medium of active ethical involvement. It gives students the chance to detect violations, report them, and observe rules being amended.

By looking at controversies and deficiencies in dealing with major issues like copyrights and copybots, students are stimulated to think of solutions to deal with these issues. This is a good way of enhancing their problem solving and analytical skills. A further enhancement into that direction could be through encouraging students to look into other virtual worlds and find out how they solve these issues.

From an academic perspective, using virtual worlds enhances the students' learning experience through engaging them in realistic situations and showing them how the different concepts work. On the long run, this improves the students' information retention as they start to refer to specific situations that happened to them during their virtual journeys and use them as examples for compliance with or violations of computer ethics. In this specific study, students' final assessments came out very positive after adopting Second Life as an e-learning tool.

4.1 Students assessments

As part of regular class activities, students evaluate the contents and methodology of teaching at the end of each semester. The following are comments made by students who participated in the second life experiments.:

- "Very creative method of teaching. "
- "Class activities were fun. Using technology in such course added another dimension to the course"

5 Conclusion

In this paper, we introduced some of our findings about testing ethical issues in virtual worlds. The outcome of our research clearly shows that virtual worlds although still miss a lot of real life regulations and laws, can be an excellent tool for creating a more dynamic learning experience.

In theoretical courses like ethics courses, there is small room for experimenting with real life situations. But, using e-learning tools like SL will allows students to get engaged in different situations of different settings, experience live violations of rules, discover bugs in laws, and even try to come up with solutions. Science students, in particular, find this type of learning more relevant to their coursework as it involves more lab experimenting.

Virtual worlds could be used for educational purposes in some areas like professional ethics and collaborative learning and software engineering.

Acknowledgments
The authors would like to thank the students of computer ethics classes at the American University of Kuwait for their input and cooperation in this paper.

References
[1] Stamatellos, Giannis. Computer Ethics a global perspective. Jones and Bartlett Publishers, 2007.

[2] Duncan Riley, "Second Life Bans Gambling Following FBI Investigation ", TechCrunch, http://techcrunch.com/2007/07/25/second-life-bans-gambling-following-fbi-investigation/ Accessed May 2011.

[3] Adam Reuters, "Linden bans CopyBot following resident protests", Second Life Reuters, http://Second life.reuters.com/stories/2006/11/15/linden-bans-copybot-following-resident-protests/ Accessed May 2011.

[4] Chris Gourlay and Abul Tahe, "Virtual jihad hits Second Life website", The Sunday Times, August 5, 2007. http://www.timesonline.co.uk/tol/news/world/middle_east/article2199193.ece Accessed May 2011.

[5] Brian Oliver, "Twenty years on, the 'hate match' between Egypt and Algeria is on again", Guardian October 10th, 2009. http://www.guardian.co.uk/football/blog/2009/oct/10/egypt-algeria-repeat-hate-match Accessed May 2011.

[6] "Second Life 'child abuse' claim" BBC NEWS, http://news.bbc.co.uk/2/hi/technology/6638331.stm Accessed May 2011.

[7] http://en.wikipedia.org/wiki/Holy_war

[8] Elizabeth Hodge, et al, The Virtual Worlds handbook, Jones and Bartlett Publishers, 2011.

[9] Janet, Lo. "Second Life: Privacy in Virtual Worlds", Office of Privacy Commissioner of Canada. http://www.priv.gc.ca/information/pub/sl_080411_e.cfm accessed May 2011.

[10] Amir Zeid, 'Using Collaborative Learning Methods in Teaching Computer Science Capstone: A Case Study,' proceedings of FECS'10 – The 2010 International Conference on Frontiers in Education: Computer Science and Computer Engineering, July 2010.

[11] Amir Zeid, "Using Creative methods for teaching professional ethics for computer science students," Proceedings of the First Kuwait Conference on e-Systems and e-Services, Kuwait, November 2009.

[12] Roderick Jones, "White Hate groups in Second Life should be studied" Meta Security, Security of Virtual Worlds. http://metasecurity.net/2008/07/08/white-hate-groups-in-second-life-should-be-studied/ Accessed May 2011.

[13] www.secondlife.com (accessed on May 15th, 2011)

Teaching Pointers in C/C++ to CS Freshman Students

C. Zhao, and F. Moinian
Department of Computing and Technology, Cameron University, Lawton, OK, USA

Abstract **-** *Pointers are one of the key features of the C and C++ languages which provide programmers flexible and powerful tools to manipulate data. Since pointers in C/C++ access data through their corresponding memory addresses, a program utilizing pointers may terminate with little useful error messages if any illegal address is accessed during run time. Experience from our teaching practice indicates that the pointer concept and its usage are difficult topics to grasp for CS undergraduate students. In this article we will discuss some advanced usages of pointers with respect to structures, functions, and objects and believe that it may be beneficial to both CS students and instructors in their teaching and learning practice.*

Keywords: pointers, structures, functions, objects, threads, processes.

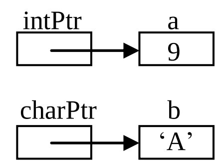
1 Introduction

Pointers are one of the most important tools in many programming languages including C/C++. They provide a mechanism to access and modify data using data's address rather than its name. Pointers are commonly used to dynamically allocate storage at run time, create lists that can grow and shrink, and pass data and functions as arguments to other functions for access and modification. Although pointers provide a powerful and flexible tool for working with different object types, they can cause a great amount of frustration to programmers, especially to freshmen computer science students. If any illegal address is accessed at runtime, the operating system will terminate the program abnormally with little or no useful error messages. Experience from our teaching practice indicates that the pointer concept and its utilization are very difficult topics for new programmers. In this paper, we intend to demonstrate some aspects of the use of pointers and present information about pointers in one document that may benefit both CS students and instructors.

1.1 Basic concepts

Pointers are variables that store addresses of other variables. Pointers may hold addresses of any type of data including predefined data and user defined data types. In C/C++ we use an asterisk before a variable name in a declaration to tell the compiler that this is a pointer. For example, we can declare a few pointers and then assign addresses of variables to them as illustrated in the following code segment.

```
int *intPtr, a;        // A pointer and an integer variable
char *charPtr, b;      // Another pointer and a character variable

intPtr = &a;
charPtr = &b;

*intPtr = 9;
*charPtr = 'A';
```



The ampersand & is the address operator. It tells the compiler that the address of a variable is to be used instead of its content. The second pair of statements in the code above store addresses of variables *a* and *b* in pointers *intPtr* and *charPtr*. In the last two lines of code the asterisk * is used as a dereference operator that tells the compiler that 9 is to be assigned to the location where *intPtr* points to, and 'A' is to be assigned to the location where *charPtr* points to. The assignment statement *intPtr = &a* makes *intPtr* and variable *a* aliases and we can use either *intPtr* or *a* to access the value 9.

Pointer variables act just like other kinds of variables. We can carry out assignment, arithmetic and comparison operations on pointers if necessary. We can also pass pointers as arguments to a function or use a pointer as a return value from a function. Since a C/C++ function may only return one value to the calling program, we can pass pointers as arguments to a function in order to fetch more than one piece of data from the called function. Note that the compiler only passes the address of the first location of a variable to the called function no matter how big the variable's memory allocation is, this avoids from creating a copy of the data and makes the program run more efficiently.

Pointers give us a powerful tool to access and modify data and communicate among functions. On the other hand, pointers can also give us some challenges. Generally speaking, pointers' syntax and usage are quite complex and difficult to grasp for beginning programmers. Any improper use of pointers in a program may cause data loss or run time errors without any warning messages. In addition, pointers do not store any useful data, and a computer system is forced to perform a "double lookup" to access a piece of data via a pointer. That is, the system looks at the address that was held by a pointer first and then accesses the value that is stored in that address. In this article we discuss some advanced usage of pointers with respect to structures, functions, and objects.

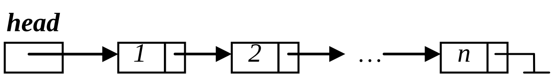
2 Pointers and Linked Lists

Linked lists provide a mechanism to store data with structures so that the programmer can automatically create a new place to store data whenever necessary. Linked lists are widely used in the system and application programming practice. Specifically, a programmer writes a structure or a class definition that contains variables holding information about some data or object, and then provides a pointer to the structure of its type. Each of these individual structures or classes in the list is commonly known as a node. Since there is at least one pointer in each node, linked lists offer a great application to learn about pointers. In order to demonstrate a simple application of pointers, we use the C syntax in our examples where necessary. The following is a singly linked list of integers in which we intend to explain how to use a pointer properly in list operations such as insertion and deletion at the beginning of the list.

A node structure contains two components, one for the value of data and the other for pointing to another node. The node structure can be defined as follows:

```
typedef struct node{
   int data;
   struct node *next;
} node_type;
```

Suppose that we want to build an integer stack by using a linked list implementation. Our stack will contain integers 1 through n and should logically look similar to the following diagram:



Assume that we write the following code in C.

```
#include <stdlib.h>
#include <stdio.h>
#define n 100

typedef struct node{
   int data;
   struct node *next;
} node_type;

void push_item(node_type *, int);
int main()
{
  node_type *head = NULL;
  int i;

  for (i = n; i > 0; i--)
    push_item(head, i);

  printf("\nThe stack contains: ");
  while (head != NULL)      // Print the stack
  {
    printf("%i ", head -> data);
    head = head -> next;
  }
  printf("\n");
  return 0;
}

void push_item(node_type *head, int data)
{
  node_type *new_node = (node_type *)
  malloc(sizeof(node_type));
  new_node -> data = data;
  new_node -> next = head;
  head = new_node;
}
```

When we compile and run this program, it generates no output! What causes this to happen? It seemed that we had a good algorithm: we allocated a piece of memory for a new node, then we assigned data to its data field and made its pointer point to the head; finally we changed the head pointer to the new node's pointer. In order to solve the run time problem we should review the concept of pointers. A pointer is a variable just like any other variable; it has its own address and content. We have to pass its own address instead of its value, which is an address of another variable, if we need to change its value. This is a very tricky area of programming in C. In our push_item function we need to be able to change the head's content which is the address of the first node of a linked list, so we have to pass to the push_item function the address of the head pointer, that is, a pointer to a pointer of a node(node_type**). When the push_item function terminates, the pointer to the head, the head pointer's address, with a new content that is the address of a new node will be returned back to the calling function, in our case the main function. Therefore we should rewrite the code as follows.

```
#include <stdlib.h>
#include <stdio.h>
#define n 100

typedef struct node{
   int data;
   struct node *next;
} node_type;

void push_item(node_type **, int);
int main()
{
  node_type *head = NULL;
  int i;

  for (i = n; i > 0; i--)
    push_item(&head, i);
```

```
  printf("\nThe stack contains: ");
  while (head != NULL)      // Print the stack
  {
    printf("%i ", head -> data);
    head = head->next;
  }
  printf("\n");
  return 0;
}

void push_item(node_type **head, int data)
{
  node_type *new_node = (node_type *)
  malloc(sizeof(node_type));
  new_node -> data = data;
  new_node -> next = *head;
  *head = new_node;
}
```

Note that NULL is only a constant pointer in C/C++ and its value is defined as 0 in <cstddef>. NULL is considered a generic pointer which means that it can be assigned to any type of pointer. Also note that an undefined pointer may still point to a random location in memory that could contain useful data. Therefore, we should assign NULL to unused and undefined pointers to avoid destroying useful data by accident.

3 Pointers and Functions

A pointer to a function is perhaps one of the most confusing cases in C. Pointers to functions are not as common as other pointer applications. However, one common use is to pass to a function a pointer as an argument in a function call.

This is especially useful when alternative functions maybe used to perform similar tasks on data. You can pass the data and the function to be used to some ***control*** function.

To declare a pointer to a function the following statement may be used.

```
int (*function_pointer)(int);
```

This simply declares a pointer called function_pointer to a function that takes an integer parameter and returns an integer data type. If we have declared the function int func(int), then we can simply write:

```
function_pointer = &func;
```

The *qsort* standard library function is a very useful function that is designed to sort an array by a key value of any type into ascending order, as long as the elements of the array are of fixed type.

The prototype of *qsort* is defined in <stdlib.h> as follows:

```
void qsort(void *base, size_t num_elements,
        size_t element_size,
        int (*compare)(void const *, void  const *));
```

There are four parameters in this declaration: the first parameter *base* points to the array to be sorted, the second parameter *num_elements* indicates how large the array is, the third parameter *element_size* is the size in bytes of each array element, and the final argument *compare* is a pointer to a function.

qsort calls the compare function which is user defined to compare the data during sorting. Since *qsort* uses void pointers as parameters, a user can use it to sort different kinds of arrays. As indicated, the compare function must return a value to indicate the result of a comparison. For example, the following code will sort an integer array of 100 random integers into non-descending order.

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#define SIZE 100

int compare (void *a, void *b)
{
  return(*((int*) a) - *((int*) b));
}

int main()
{
  int arr[SIZE], i;
  int (*fp)(void*, void*);
  unsigned int seed;

  seed = time(NULL);
  srand(seed);
  fp = &compare;

  for(i = 0; i < SIZE; i++)   // Fill the array with random
    arr[i] = rand() % 20;     // numbers.

  qsort(arr, SIZE, sizeof(int), fp);
  printf("\n");

  for(i = 0; i < SIZE; i++)  // Print the sorted array
    printf( " %d ", arr[i]);

  printf("\n");
  return 0;
}
```

We can use this technique to sort a complex structure array according to a sort key as long as we rewrite the compare function.

Another example of pointer to a function is in thread applications. A thread is a basic unit of CPU utilization; it has a program counter, a register set, and a stack. All threads in the same process share its code section, data section, open files, memory, and other resources. If a process has multiple threads of control, it can perform more than one task at a time and therefore, it may improve system performance significantly. If a new thread is created in a process, we need to load a task into it to complete. The following is an example of such usage. We create a thread first, then, we load a task that calculates the value of the Fibonacci sequence at a specified position.

```
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>

void* fibonacci(void*);

int main(int argc, char *argv[])
{
  void (*func_pointer)(void*);
  pthread_t tid;
  pthread_attr_t attr;

  if(argc != 2)
  {
    fprintf(stderr, "usage: execFile PositiveInt\n");
    exit(1);
  }

  func_pointer = (void*) &fibonacci;
  pthread_attr_init(&attr);
  pthread_create(&tid, &attr, (void*) func_pointer, argv[1]);
  pthread_join(tid, NULL);
  exit(0);
}

void* fibonacci(void* ptr)
{
  int i, pos, cur, first, second;

  pos = atoi((char*) ptr);
  if (pos == 0 || pos == 1)
    printf("Value at position %d of the Fibonacci sequence is:
             %d\n", pos, pos);
  else {
    first = 0;
    second = 1;
    for (i = 2; i <= pos; i++)
    {
     cur = first + second;
     first = second;
     second = cur;
    }
    printf("Value at position %d of the Fibonacci sequence is:
            %d\n", pos, cur);
  }
}
```

4 Pointers in Polymorphism and Dynamic Binding

In object-oriented programming polymorphism refers to a programming language's ability to process objects differently depending on their data type or class. More specifically, it is the ability to redefine methods for derived classes. One of the key features of derived classes is that a pointer to a derived class is type-compatible with a pointer to its base class. Polymorphism takes advantage of this simple but powerful and versatile feature, which brings Object Oriented Methodologies to its full potential.

For example, given a base class *shape*, polymorphism enables the programmer to define different *area* methods for any number of derived classes such as circles, rectangles and triangles. No matter what shape an object is, applying the *area* method to it will return the correct result. Polymorphism is considered to be a requirement of any true object-oriented programming language. In C++ both general and specialized methods need to be virtual methods.

In the following C++ program we show how this feature can be implemented.

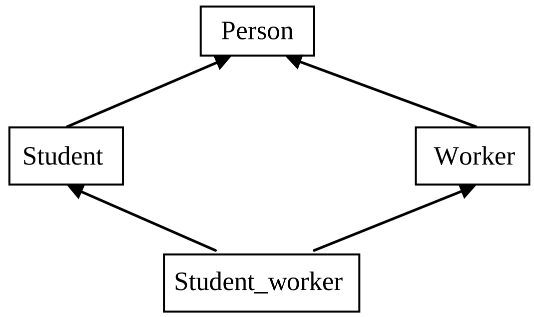
```
#include <iostream>
using namespace std;

class person
{
 public:
   person(char *str)
  {
    strcpy(name, str);
  }
 virtual void print_info()
 {
   cout << "My name is " << name << endl;
 }
 protected:
  char name[20];
};

class student:virtual public person
{
 public:
   student(char *str, float g):person(str), gpa(g){}
   void print_info()
   {
    person::print_info();
    cout << "My GPA is " << gpa << endl;
   }
 protected:
  float gpa;
};
```

```
class worker: virtual public person
{
  public:
    worker(char *str, int h):person(str), hours(h){}
    void print_infor()
    {
      person::print_info();
      cout << "I work " << hours << " hours" << endl;
    }
  protected:
    int hours;
};

class student_worker:public student, worker
{
  public:
    student_worker(char *str, float g, int h):person(str),
    student(str,g), worker(str,h){}
    void print_info()
    {
      cout << "My name is " << name << endl;
      cout << "My GPA  is " << gpa  << endl;
      cout << "I work " << hours << " hours" << endl;
    }
};

int main()
{
  person me ("Maxwell");
  student you ("Wang", 3.0);
  worker he ("Wang", 40);
  student_worker she ("Mary", 3.9, 40);
  person *p;

  p = &me;
  p -> print_info();

  p = &you;
  p -> print_info();

  p = &he;
  p -> print_info();

  p = &she;
  p -> print_info();
  return 0;
}
```

The program will generate the following output:

```
My name is Maxwell
My name is Wang
My GPA is 3
My name is Wang
My name is Mary
My GPA is 3.9
I work 40 hours
```

In the example above we established the following hierarchical inheritance.



Polymorphism allows all classes to use the same method, namely *print_info*, to print their own message. The program performance may be improved since dynamic binding technique is used here. This illustrates that while complex, pointers are very important concepts in object oriented programming paradigm.

5 Conclusions

This paper presented some elementary and advanced applications of using pointers in C/C++. Pointers allow programmers to request memory for storing data on a need basis at run time. They provide an efficient mechanism to pass and return large volumes of data to and from functions. Pointers also allow functions to be passed to functions as parameters, and play an important role in working with polymorphic classes. Teaching these concepts to freshman students in CS remains a challenging task. Some of the difficulties involve the abstraction level of the concept of pointers along with the lack of sufficient run time support for generating helpful error messages. It is our recommendation that a first course in C/C++ for CS freshman students should not include the coverage of pointers. The second course can introduce pointers and illustrate its use in linked lists and pointer parameters. Finally, the third course may cover the object oriented methodology along with inheritance, polymorphism and the advanced applications of pointers.

6 References

[1] Abraham Silberschatz, Greg Gagne, Peter B. Galvin. "Operating System Concepts, 7th edition". John Wiley & Sons, 2005.

[2] Frank M. Carrano. "Data Abstraction and Problem Solving with C++ Walls and Mirrors, 4th edition". Addison-Wesley, 2005.

[3] Larry Nyhoff. "C++ An Introduction to Data Structures". Prentice Hall, 1999.

[4] Jeri R. Hanly, Elliot B. Koffman. "Problem Solving and Program Design". Addison-Wesley, 2004.

SESSION

CASE STUDIES AND PROJECTS

Chair(s)

TBA

An Iterative Scheme for Numerical Solution of Nonlinear Integro-differential Equations

K. Maleknejad, E. Najafi
School of Mathematics, Iran University of Science and Technology, Narmak, Tehran 16846 13114, Iran

Abstract— *The method of generalized quasilinearization technique when is applied to the nonlinear integro-differential equations of Volterra type, gives two sequences of linear integro-differential equations with solutions monotonically and quadratically convergent to the solution of nonlinear equation. In this paper we employ step-by-step collocation method to solve the linear equations numerically and then approximate the solution of the nonlinear equation. Error analysis is performed and some numerical examples are given that observe the accuracy of the method.*

Keywords: Integro-differential equations, Collocation method, Quasilinearization technique.

1. Introduction

It is well known that the method of quasilinearization [1] provides an excellent tool for obtaining approximated solutions of nonlinear differential equations. This technique works fruitfully for the problems that their nonlinear parts involve convex or concave functions and gives two sequences of linear problems that their solutions are upper and lower solutions to the nonlinear problem and are converging monotonically and quadratically to the unique solution of the given nonlinear problem. Recently, this method is applied to a variety of problems [2–5] and in the continuation the convexity assumption was relaxed and the method was generalized and extended in various directions to make it applicable to a large class of problems [6–14].

The method of quasilinearization is an effective tool to obtain lower or upper bounds for the solutions of nonlinear differential equations [1,15]. To describe it, consider the initial-value problem (IVP)

$$x'(t) = f(t, x(t)), \qquad x(0) = x_0, \tag{1}$$

on $J = [0, T]$. If f is convex on x, then one can find a function $g(t, x, y)$ which is linear in x such that

$$f(t, x) = \max_y g(t, x, y).$$

By choosing an initial approximation $y_0(t)$, using $g(t, x, y)$ one can generate a monotone sequence $y_n(t)$, that converges quadratically to the unique solution of Eq. (1). Moreover, the sequence provides good lower bounds for the solution. If f is concave on x, on the other hand, a dual result holds that offers monotone approximations having similar properties and good upper bounds.

In the area of integro-differential equations consider the IVP

$$x'(t) = f(t, x(t)) + \int_{t_0}^{t} k(t, s, x(s))ds, \qquad x(t_0) = x_0, \tag{2}$$

with

$$t \in J = [t_0, t_0 + T], \quad , t_0 \geq 0, \quad T > 0.$$

When the numerical methods are applied to solve these nonlinear problems, they mostly are converted to a nonlinear algebraic system. Eq. (2) is studied in [16] using multistep rules with quadrature formulas and in [17] using collocation method and in both of them the integral terms are discretized to nonlinear algebraic systems. But these nonlinear systems need some conditions to have a unique solution and require an iteration method (In many cases the Newton's iteration) and a suitable starting point to converge to the solution. When one use the multistep methods, the process of solving these nonlinear algebraic systems is repeated in each step to acquire the next step nodes and this process causes a lot of computational costs and additional works. In Eq. (2) when f and k are convex in x, the method of quasilinearization [12] is applicable and offers the following two linear iterative schemes

$$\alpha'_{p+1}(t) = f(t, \alpha_p(t)) + f_x(t, \alpha_p(t))(\alpha_{p+1}(t) - \alpha_p(t))$$
$$+ \int_{t_0}^{t} \Big[k(t, s, \alpha_p(s)) + k_x(t, s, \alpha_p(s))(\alpha_{p+1}(s) - \alpha_p(s))\Big] ds,$$
$$\alpha_{p+1}(t_0) = x_0, \tag{3}$$

$$\beta'_{p+1}(t) = f(t, \beta_p(t)) + f_x(t, \alpha_p(t))(\beta_{p+1}(t) - \beta_p(t))$$
$$+ \int_{t_0}^{t} \Big[k(t, s, \beta_p(s)) + k_x(t, s, \alpha_p(s))(\beta_{p+1}(s) - \beta_p(s))\Big] ds,$$
$$\beta_{p+1}(t_0) = x_0, \tag{4}$$

for $p = 0, 1, 2, \ldots$, as two linear integro-differential equations, where $\alpha_0(t)$ and $\beta_0(t)$ are the upper and lower solutions of Eq. (2), presented in below. The solutions of the iterative schemes (3) and (4) are quadratically and monotonically convergent to the unique solution of (3). In this paper we apply step-by-step collocation method in a piecewise continuous polynomials space to solve numerically the linear equations (3) and (4). We combine this and the iterative schemes (3) and (4) (where with respect to their linearity and quadratically convergent is rapid in convergence) to approximate the unique solution of Eq. (2). This paper has been organized as follows. A general framework of the idea of quasilinearization used to solve the nonlinear integro-differential equations and some conclusions are recalled in section 2. Section 3 shows employing step-by-step collocation method in approximating the solution of the linear integro-differential equations in a piecewise continuous polynomials space. Section 4 includes discretization to a linear algebraic system and discussion of the convergence of the method and in section 5, the suggested method is applied on some numerical examples.

2. Integro-differential inequalities and quasilinearization

Consider the nonlinear integro-differential equation

$$x'(t) = f(t, x(t)) + \int_0^t k(t, s, x(s))ds, \qquad x(0) = x_0, \quad (5)$$

for $f \in C[J \times \mathbb{R}, \mathbb{R}]$ and $k \in C[\mathcal{D} \times \mathbb{R}, \mathbb{R}]$ where $T \in \mathbb{R}$ and $t \in J = [0, T]$ and $\mathcal{D} = \{(t, s) \in J \times J : s \leq t\}$. Using the mapping $t \longrightarrow t - t_0$ Eq.(2) is reducible to Eq.(5).

Definition 1: The function $\alpha \in C^1[J, \mathbb{R}]$ is said to be a lower solution for Eq.(5) on J if

$$\alpha'(t) \leq f(t, \alpha(t)) + \int_0^t k(t, s, \alpha(s))ds, \qquad \alpha(0) \leq x_0,$$

and an upper solution if the reversed inequality is satisfied. We have the following conclusions about lower and upper solutions of the IVP (5) where are expressed in [12].

Lemma 1: Consider the IVP (5) and that:
$(\mathbf{A_1})$ $f \in C[J \times \mathbb{R}, \mathbb{R}]$ and $k \in C[\mathcal{D} \times \mathbb{R}, \mathbb{R}]$ and $k(t, s, x)$ is monotone nondecreasing in x for each fixed $(t, s) \in \mathcal{D}$;
$(\mathbf{A_2})$ $\alpha_0(t), \beta_0(t) \in C^1[J, \mathbb{R}]$ are lower and upper solutions of the IVP (5) respectively;
$(\mathbf{A_3})$ for $(t, s) \in \mathcal{D}$, $v \geq w$, $L \geq 0$, $f(t, v) - f(t, w) \leq L(v - w)$, and $k(t, s, v) - k(t, s, w) \leq N(v - w)$, $N > 0$.
Then we have $\alpha_0(t) \leq \beta_0(t)$ for $t \in J$, provided $\alpha_0(t_0) \leq \beta_0(t_0)$.
If the Lemma (1) holds, it is shown that the IVP (5) has a unique solution $x(t)$ such that satisfies in the relation

$$\alpha_0(t) \leq x(t) \leq \beta_0(t), \qquad t \in J.$$

Using the norm $\|x\| = \max_{t \in J} |x(t)|$ and defining two iterative schemes (3) and (4) the following theorem is applied for the unique solution of (5):

Theorem 1: Assume that:
$(\mathbf{B_1})$ $f \in C^2[J \times \mathbb{R}, \mathbb{R}]$, $k \in C^2[\mathcal{D} \times \mathbb{R}, \mathbb{R}]$ and $\alpha_0(t), \beta_0(t) \in C^1[J, \mathbb{R}]$ are lower and upper solutions of the IVP (5) such that $\alpha_0(t) \leq \beta_0(t)$ on J;
$(\mathbf{B_2})$ $f_{xx}(t, x) \geq 0$ for each $t \in J$ and $k_{xx}(t, s, x) \geq 0$ for each $(t, s) \in \mathcal{D}$;
$(\mathbf{B_3})$ $k(t, s, x)$ is monotone nondecreasing in x for each $(t, s) \in \mathcal{D}$ and for each $\alpha_0(t) \leq v(t) \leq \beta_0(t)$.

Then the monotone sequences $\{\alpha_p(t)\}$ and $\{\beta_p(t)\}$ generated by iterative schemes (3) and (4) converge uniformly and quadratically to the unique solution of (5) on J,

$$\begin{aligned} \|x - \alpha_{p+1}\| &\leq \frac{2e^{LT}}{\sqrt{L^2 + 4L_1}}(M + M_1)\|x - \alpha_p\|^2, \\ \|\beta_{p+1} - x\| &\leq \frac{2e^{LT}}{\sqrt{L^2 + 4L_1}}(M + M_1)\|\beta_p - x\|^2, \\ L &= \max_{t \in J} |f_x(t, x)|, \qquad L_1 = \max_{t \in J} |k_x(t, s, x)|, \\ M &= \max_{t \in J} f_{xx}(t, x), \qquad M_1 = \max_{t \in J} k_{xx}(t, s, x), \end{aligned}$$

and satisfy the relation

$$\alpha_0(t) \leq \alpha_1(t) \leq \cdots \leq \alpha_p(t) \leq \beta_p(t) \leq \cdots \leq \beta_1(t) \leq \beta_0(t).$$

The following Lemma in [18] is required to establish the convergence of the presented method.

Lemma 2: Suppose $\|.\|$ is a subordinate matrix norm for which the norm of the identity matrix $\|I\| = 1$ and E is a matrix such that $\|E\| < 1$. Then $(I - E)$ is nonsingular and

$$\|(I - E)^{-1}\| \leq (1 - \|E\|)^{-1}.$$

3. Piecewise Polynomials Collocation Method

We define the partition $\{0 = t_0 < t_1 < \cdots < t_N = T\}$ on J, $h_n = (t_{n+1} - t_n)$, $n = 0, \cdots, N - 1$, $h = \max_n\{h_n\}$, and indicate the above partition by J_h. According to this partition we have the following definition.

Definition 2: Suppose that J_h is a given partition on J. The piecewise polynomials space $S_m^{(d)}(J_h)$ with $m \geq 0$, $-1 \leq d \leq m$ is defined by

$$S_m^{(d)}(J_h) = \{q(t) \in C^d[J, \mathbb{R}] : q|_{\sigma_n} \in \pi_m \;; 0 \leq n \leq N - 1\}.$$

Here $\sigma_n = (t_n, t_{n+1}]$ and π_m denotes the space of polynomials of degree not exceeding m. Now, consider the linear integro-differential equation (3). We may show it in the form of

$$\alpha_p'(t) = H_p(t) + f_p(t)\alpha_p(t) + \int_0^t k_p(t, s)\alpha_p(s)ds, \qquad \alpha_p(0) = x_0, \quad (6)$$

where

$$H_p(t) = f(t, \alpha_{p-1}(t)) - f_x(t, \alpha_{p-1}(t))\alpha_{p-1}(t) + \int_0^t \Big(k(t, s, \alpha_{p-1}(s)) - k_x(t, s, \alpha_{p-1}(s))\alpha_{p-1}(s)\Big)ds, \quad (7)$$

and

$$f_p(t) = f_x(t, \alpha_{p-1}(t)), \qquad k_p(t, s) = k_x(t, s, \alpha_{p-1}(s)). \quad (8)$$

We approximate the solution of the linear IVP (3) in the continuous polynomials space

$S_m^{(0)}(J_h)$={$q(t) \in C^d[J,\mathbb{R}]$: $q|_{\sigma_n} \in \pi_{m-1}$; $0 \le n \le N-1$},

by collocation method corresponding to the choice $d=0$ in Definition (2) The collocation solution is denoted by $\hat{\alpha}_p(t)$ and defined by the collocation equation

$$\hat{\alpha}'_p(t) = H_p(t) + f_p(t)\hat{\alpha}_p(t) + \int_0^t k_p(t,s)\hat{\alpha}_p(s)ds, \quad \hat{\alpha}_p(0) = x_0, \quad t \in X_h, \quad (9)$$

where X_h contains the collocation points

$$X_h = \{t_n + c_i h_n : 0 \le c_1 \le \ldots \le c_m \le 1; 0 \le n \le N-1\}, \quad (10)$$

and is determined by the points of the partition J_h and the given collocation parameters $\{c_i\} \in [0,1]$.

4. Lagrange Basis Functions and Discretization

A comfort computational form of the collocation equation is obtained when the Lagrange basis functions are used to form a basis in each subinterval σ_n for the space $S_m^{(0)}(J_h)$ with respect to the collocation parameters $\{c_i\}$.
These polynomials in σ_n can be written as

$$L_j(z) = \prod_{k \ne j}^{m} \frac{z - c_k}{c_j - c_k}, \qquad z \in [0,1], \quad j = 1, \cdots, m, \quad (11)$$

where belong to π_{m-1}. Also set

$$A^p_{n,j} = \hat{\alpha}'_p(t_n + c_j h_n), \qquad j = 1, \cdots, m. \quad (12)$$

The restriction of the collocation solution $\hat{\alpha}_p(t) \in S_m^{(0)}(J_h)$ to the subinterval σ_n is as

$$\hat{\alpha}'_{p,n}(z) = \hat{\alpha}'_p(t_n + zh_n) = \sum_{j=1}^{m} L_j(z) A^p_{n,j}, \quad z \in (0,1], \quad (13)$$

and by letting $\mathcal{L}_j(z) = \int_0^z L_j(v)dv$ and $y_n = \hat{\alpha}_p(t_n)$ we obtain from (13) the representation of $\hat{\alpha}_p(t)$ on σ_n as follows:

$$\begin{aligned}\hat{\alpha}_{p,n}(z) &= \hat{\alpha}_p(t_n + zh_n) \\ &= y_n + h_n \sum_{j=1}^{m} \mathcal{L}_j(z) A^p_{n,j}, \qquad z \in (0,1]. \quad (14)\end{aligned}$$

Now, for $t = t_{n,i} = t_n + c_i h_n$, the collocation equation (9) has the following form:

$$\hat{\alpha}'_p(t_{n,i}) = H_p(t_{n,i}) + f_p(t_{n,i})\hat{\alpha}_p(t_{n,i}) + \int_0^{t_n} k_p(t_{n,i},s)\hat{\alpha}_p(s)ds + h_n \int_0^{c_i} k_p(t_{n,i}, t_n + sh_n)\hat{\alpha}_p(t_n + sh_n)ds, \quad \hat{\alpha}_p(0) = x_0.$$

By using (12), (13) and (14), it is written as follows:

$$\begin{aligned}A^p_{n,i} &- h_n f_p(t_{n,i}) \sum_{j=1}^{m} \mathcal{L}_j(c_i) A^p_{n,j} \\ &- h_n^2 \sum_{j=1}^{m} \Big(\int_0^{c_i} k_p(t_{n,i}, t_n + sh_n)\mathcal{L}_j(s)ds \Big) A^p_{n,j} \\ &= H_p(t_{n,i}) + F^n_p(t_{n,i}) \\ &+ \Big(f_p(t_{n,i}) + h_n \int_0^{c_i} k_p(t_{n,i}, t_n + sh_n)ds \Big) y_n, \quad (15)\end{aligned}$$

for $i = 1, \ldots, n$, where

$$\begin{aligned}F^n_p(t) &= \int_0^{t_n} k_p(t,s)\hat{\alpha}_p(s)ds \\ &= \sum_{\ell=0}^{n-1} h_\ell \int_0^1 k_p(t, t_\ell + sh_\ell)\hat{\alpha}_p(t_\ell + sh_\ell)ds, \quad (16)\end{aligned}$$

denotes the collocation solution on $[0, t_n]$. Using representation (14) and setting $t = t_{n,i}$ in (16) deduce

$$\begin{aligned}F^n_p(t_{n,i}) &= \sum_{\ell=0}^{n-1} h_\ell \Big(\int_0^1 k_p(t_{n,i}, t_\ell + sh_\ell)ds \Big) y_\ell \\ &+ \sum_{\ell=0}^{n-1} h_\ell^2 \sum_{j=1}^{m} \Big(\int_0^1 k_p(t_{n,i}, t_\ell + sh_\ell)\mathcal{L}_j(s)ds \Big) A^p_{\ell,j},\end{aligned}$$

and by defining

$$B^{\ell,n}_p = \begin{pmatrix} \int_0^1 k_p(t_{n,i}, t_\ell + sh_\ell)\mathcal{L}_j(s)ds \\ i,j = 1, \cdots, m \end{pmatrix},$$

for $0 \le \ell < n \le N-1$, as a $(m \times m)$ matrix, has the form

$$\begin{aligned}F^n_p(t_{n,i}) &= \sum_{\ell=0}^{n-1} h_\ell \Big(\int_0^1 k_p(t_{n,i}, t_\ell + sh_\ell)ds \Big) y_\ell \\ &+ \sum_{\ell=0}^{n-1} h_\ell^2 \sum_{j=1}^{m} \Big[B^{\ell,n}_p \Big]_{ij} A^p_{\ell,j}, \qquad i = 1, \ldots, m,\end{aligned}$$

where $\Big[B^{\ell,n}_p \Big]_{ij}$ shows the (i,j)'th component of the matrix $B^{\ell,n}_p$. By letting

$$\begin{aligned}C^{\ell,n}_p &= \Big(\int_0^1 k_p(t_{n,1}, t_\ell + sh_\ell), \ldots, \int_0^1 k_p(t_{n,m}, t_\ell + sh_\ell) \Big)^T, \\ C^n_p &= \Big(\int_0^{c_1} k_p(t_{n,1}, t_n + sh_n), \ldots, \int_0^{c_m} k_p(t_{n,m}, t_n + sh_n) \Big)^T, \\ A^n_p &= \big(A^p_{n,1}, \ldots, A^p_{n,m} \big)^T, \\ H^n_p &= \big(H_p(t_{n,1}), \ldots, H_p(t_{n,m}) \big)^T, \\ f^n_p &= \big(f_p(t_{n,1}), \ldots, f_p(t_{n,m}) \big)^T, \\ G^n_p &= \big(F^n_p(t_{n,1}), \ldots, F^n_p(t_{n,m}) \big)^T,\end{aligned}$$

and defining the $(m \times m)$ matrices

$$L_p^n = diag\Big(f_p(t_{n,i})\Big)\left(\begin{array}{c}\mathcal{L}_j(c_i)\\ i,j=1,\dots,m\end{array}\right),$$
$$B_p^n = \left(\begin{array}{c}\int_0^{c_i} k_p(t_{n,i}, t_n + sh_n)\mathcal{L}_j(s)ds\\ i,j=1,\dots,m\end{array}\right),$$

for $0 \le n \le N-1$, the collocation equation (9) is reduced to the linear algebraic system

$$\Big(I_m - h_n(L_p^n + h_n B_p^n)\Big)A_p^n = H_p^n + G_p^n + (f_p^n + h_n C_p^n)y_n, \quad 0 \le n \le N-1, \quad p = 1, 2, \cdots. \tag{17}$$

Here I_m denotes the $(m \times m)$ identity matrix. The existence and uniqueness of the collocation solution in $S_m^{(0)}(J_h)$ is considered in the following theorem.

Theorem 2: If $H_p(t)$, $f_p(t)$ and $k_p(t,s)$ in the Volterra integro-differential equation (9) are all continuous on their domains J and D, then there exists an $\bar{h} > 0$ such that for any partition J_h with partition diameter h, $0 < h < \bar{h}$, the linear algebraic system (17) has a unique solution A_p^n for $0 \le n \le N-1$ and $p = 1, 2, \cdots$.

Proof. The continuity of $H_p(t)$, $f_p(t)$ and $k_p(t,s)$ is obvious with respect to (7), (8) and Theorem (1). On the other hand the domains of $f_p(t)$ and $k_p(t,s)$ are compact, then the components of the matrices L_p^n and B_p^n for $0 \le n \le N-1$ and $p = 1, 2, \cdots$, are all bounded. These implies if h_n's are chosen sufficiently small, the inequality $h_n\|L_p^n + h_n B_p^n\| < 1$ holds and by Lemma (2) the inverse of the matrix $(I_m - h_n(L_p^n + h_n B_p^n)$ exists. In other words, there is an $\bar{h} > 0$ so that for any partition J_h with $h = \max\{h_n;\ 0 \le n \le N-1\} < \bar{h}$ the matrix $I_m - h_n(L_p^n + h_n B_p^n)$ has a uniformly bounded inverse and the proof is complete. ■

When the unknown vector A_p^n is computed from (17), the collocation solution for $t = t_n + zh_n \in \bar{\sigma}_n = [t_n, t_{n+1}]$ is given by

$$\hat{\alpha}_p(t_n + zh_n) = y_n + h_n \sum_{j=1}^{m} \mathcal{L}_j(z)A_{n,j}^p, \qquad z \in (0,1].$$

The following theorem shows that this collocation solution is convergent to the solution of linear integro-differential equation (6). The proof of this theorem with some changes may be found in [17].

Theorem 3: Suppose that in (6) $k_p(t,s) \in C^i[D,\mathbb{R}]$ and $H_p(t), f_p(t) \in C^i[J,\mathbb{R}]$, where $1 \le i \le m$, and $\hat{\alpha}_p \in S_m^{(0)}(J_h)$ is the collocation solution of equation (6) with $h \in (0,\bar{h})$. If $\alpha_p(t)$ is the exact solution of equation (6), then

$$\|\alpha_p - \hat{\alpha}_p\| \le C\|\alpha_p^{(i+1)}\|h^i,$$

holds on J, for any collocation points X_h. The constant C depends on the parameters $\{c_i\}$ but not on h.

The above argument yields an approximation solution $\hat{\alpha}_p(t)$ to the unique solution of the linear integro-differential equation (6) in the space $S_m^{(0)}(J_h)$ and the iterative scheme (3) or (6) produces a sequence $\{\alpha_p(t)\}$ that is quadratically convergent to the unique solution of nonlinear integro-differential equation (5). The inequality

$$\|x - \hat{\alpha}_p\| \le \|x - \alpha_p\| + \|\alpha_p - \hat{\alpha}_p\|, \tag{18}$$

and Theorems (1) and (3) show that the sequence of the collocation solutions $\{\hat{\alpha}_p(t)\}$ is convergent to the unique solution $x(t)$ of nonlinear equation (5). It is noticeable that in the relation (18) the first term is quadratically convergent and the convergence of the second term is $O(h^m)$.

5. Numerical Experiments

For numerical experiments, the presented method is applied to solve three different examples of the integro-differential equation (5). In each three examples the subintervals and the collocation parameters are chosen such that

$$h = h_n = \frac{T}{N}, \qquad n = 0, \dots, N-1,$$
$$c_i = \frac{i-1}{m-1}, \qquad i = 1, \dots, m.$$

Example 1: The first example is the following integro-differential equation:

$$x'(t) = x^5(t) + 1 - \frac{5}{4}t^5 + \int_0^t tsx^2(s)ds, \quad x(0) = 0, \tag{19}$$

where $0 \le t \le 1$, $k(t,s,x) = tsx^2$ and $f(t,x) = x^5 + 1 - \frac{5}{4}t^5$. Then, k is nondecreasing and k and f are both convex with respect to x on $\mathcal{D}$ and $\alpha_0(t) = \frac{t}{2}$ is a lower solution of (19) on $[0,1]$. On the other hand, the exact solution is $x(t) = t$ where with respect to the Theorem (1) the solutions of the iterative scheme

$$\alpha_p'(t) = H_p(t) + 5\alpha_{p-1}^4(t)\alpha_p(t) + 2\int_0^t ts\alpha_{p-1}(s)\alpha_p(s)ds,$$
$$\alpha_p(0) = 0, \qquad p = 1, 2, \dots,$$

with

$$H_p(t) = 1 - \frac{5}{4}t^5 - 4\alpha_{p-1}^5(t) - \int_0^t ts\alpha_{p-1}^2(s)ds,$$

is convergent to the exact solution of (19). To employ the given numerical procedure for approximating the solutions of these linear integro-differential equations, the values $m = 4, N = 5$ are chosen. The absolute values of errors, $|x(t_i) - \hat{\alpha}_p(t_i)|$, for equation (19) are shown in Table 1 and Figure 1 shows the convergence of the sequence $\{\hat{\alpha}_p(t)\}$ to the exact solution.

Table 1: Absolute errors in Example 1: $|x(t_i) - \hat{\alpha}_p(t_i)|$.

$m = 4$, $N = 5$			
t_i	$p = 2$	$p = 4$	$p = 6$
0.125	2.5576 E -12	5.551115 E -15	5.5511 E -17
0.250	1.0678 E -11	7.216450 E -14	1.1102 E -17
0.375	6.5008 E -08	8.159384 E -14	2.2493 E -16
0.500	2.5310 E -07	9.477908 E -13	2.8270 E -16
0.625	1.8806 E -05	6.283990 E -12	3.0340 E -16
0.750	1.1042 E -04	2.160307 E -12	5.6564 E -15
0.875	1.4790 E -04	1.491680 E -11	7.1957 E -15
1.000	9.4680 E -03	3.598937 E -10	7.1534 E -15

Table 2: Absolute errors in Example 2 for lower solutions: $|x(t_i) - \hat{\alpha}_p(t_i)|$.

	$m = 4$, $N = 5$			$m = 6$, $N = 5$		
t_i	$p = 2$	$p = 4$	$p = 6$	$p = 2$	$p = 4$	$p = 6$
0.125	9.9528 E -10	9.9528 E -10	9.9528 E -10	1.8496 E -13	1.8995 E -13	1.8995 E -13
0.250	9.9683 E -09	4.9761 E -09	4.9761 E -09	2.5706 E -09	1.0480 E -12	1.0480 E -12
0.375	2.3559 E -09	1.4969 E -08	9.2451 E -09	1.3277 E -08	7.9847 E -13	7.9847 E -13
0.500	4.3242 E -07	2.0257 E -08	2.0257 E -08	5.5749 E -07	2.3450 E -12	2.3450 E -12
0.625	1.5111 E -06	2.8473 E -08	2.8473 E -08	1.5322 E -06	3.5107 E -12	3.5107 E -12
0.750	3.6688 E -05	7.0239 E -08	7.0248 E -08	3.2160 E -05	2.0534 E -11	5.1030 E -12
0.825	3.2453 E -04	1.1222 E -07	1.0102 E -07	2.1509 E -04	8.8511 E -10	8.2789 E -12
1.000	9.8331 E -04	1.2696 E -07	1.3180 E -07	7.6373 E -04	3.7158 E -09	7.9523 E -12

Table 3: Absolute errors in Example 3 for lower solutions: $|x(t_i) - \hat{\alpha}_p(t_i)|$.

	$m = 5$, $N = 5$			$m = 6$, $N = 5$		
t_i	$p = 2$	$p = 4$	$p = 6$	$p = 2$	$p = 4$	$p = 6$
0.125	1.3983 E -10	1.3983 E -10	1.3983 E -10	1.7406 E -12	1.7406 E -12	1.7406 E -12
0.250	1.7805 E -10	1.7805 E -10	1.7805 E -10	4.6023 E -11	4.6027 E -12	4.6027 E -12
0.375	1.6270 E -10	1.6270 E -10	1.6270 E -10	3.3139 E -11	3.3147 E -12	3.3147 E -12
0.500	5.9400 E -11	3.4832 E -11	3.4832 E -11	2.2047 E -10	5.1262 E -11	5.1262 E -12
0.625	1.8052 E -11	8.1493 E -11	8.1493 E -11	1.0421 E -10	5.3321 E -11	5.3320 E -12
0.750	3.4455 E -08	5.1929 E -11	5.1929 E -11	2.4242 E -08	4.8154 E -11	4.8152 E -12
0.825	5.6256 E -07	2.1977 E -11	2.1977 E -11	4.5155 E -07	4.8538 E -10	4.8533 E -12
1.000	3.0593 E -06	9.6862 E -11	9.6863 E -11	2.2083 E -06	2.6134 E -10	2.6129 E -11

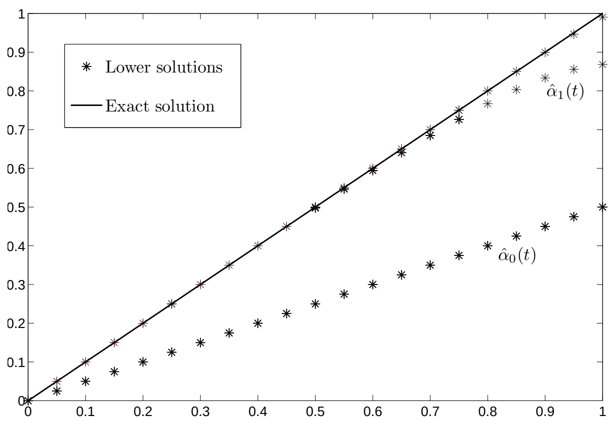


Figure 1: Convergence of the sequence $\{\hat{\alpha}_p(t)\}$ to the exact solution of Example 1.

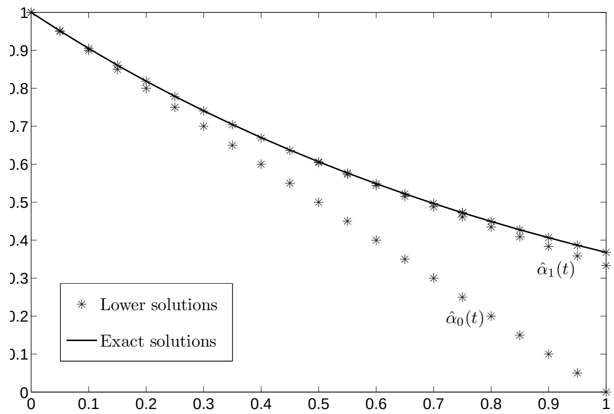


Figure 3: Convergence of the sequence $\{\hat{\alpha}_p(t)\}$ to the exact solution of Example 3.

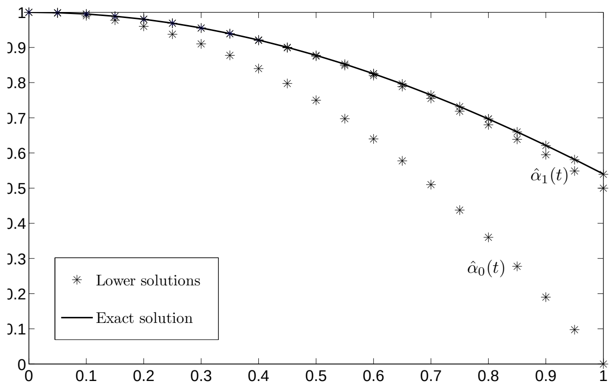


Figure 2: Convergence of the sequence $\{\hat{\alpha}_p(t)\}$ to the exact solution of Example 2.

Example 2: The second example,

$$x'(t) = \frac{t}{5}(x^5(t) - 1) - sin(t) + \int_0^t tsin(s)x^4(s)ds,$$

with initial condition $x(0) = 1$ has the exact solution $x(t) = cos(t)$ and the lower solution $\alpha_0(t) = 1 - t^2$ in the interval $[0, 1]$. The kernel of this equation satisfies the necessary conditions and as in example 1 the method is applied to this equation and the absolute values of the errors are presented in Table 2 and Figure 2.

Example 3: As the third example, we have examined the method on the problem

$$x'(t) = t^2 - (1 + t^2 + t^3)x(s) + sin(t^2) - sin(t^2x(t)) - \int_0^t t^2x(s)\big(s + cos(t^2x(s))\big)ds, \quad x(0) = 1,$$

for $0 \leq t \leq 1$, with the exact solution $x(t) = e^{-t}$. This problem is contained in the assumptions of the presented method with the lower solution $\alpha_0(t) = 1 - t$. Table 3 and Figure 3 show obtained results about this problem.

6. Conclusions

In this article we applied the method of quasilinearization and approximated the solution of nonlinear Volterra integro-differential equation. Collocation method was employed to solve the arisen linear integro-differential equations. Of advantage of the presented method is that we do not encounter solving nonlinear algebraic systems. Obtained numerical results show the accuracy and efficiency of the method. A weakness for this method is its limiting assumptions that a nonlinear equation must have them to be solvable by this method.

References

[1] R. Bellman, R. Kalaba, Quasilinearization and Nonlinear Boundary Value Problems, Elsevier, New York, 1965.

[2] J. Jiang, A.S. Vatsala, The quasilinearization method in the system of reaction diffusion equations, Appl. Math. Comput. 97 (1998) 223-235.

[3] J.I. Ramos, Piecewise-quasilinearization techniques for singularly perturbed Volterra integro-differential equations, Appl. Math. Comput. 188 (2007) 1221–1233.

[4] B. Ahmad, A quasilinearization method for a class of integro-differential equations with mixed nonlinearities, Nonlinear Anal: Real World Appl. 7 (2006) 997–1004.

[5] S.G. Pandit, Quadratically converging iterative schemes for nonlinear volterra integral equations and an application, J.AMSA, 10 (1997), 169-178.

[6] P.W. Eloe, Y. Zhang, A quadratic monotone iteration scheme for two-point boundary value problems for ordinary differential equations, Nonlinear Anal. 33 (1988) 443–453.

[7] K. Maleknejad, E. Najafi, Numerical solution of nonlinear Volterra integral equations using the idea of quasilinearization, Commun. Nonlinear. Sci. Numer. Simulat. 16 (2011) 93–100.

[8] K. Maleknejad, H. Almasieh, Optimal control of Volterra integral equations via triangular functions, Mathematical and Computer Modelling, In Press.

[9] V. Lakshmikantham, An extension of the method of quasilinearization, J. Optim. Theory Appl. 82 (1994) 315–321.

[10] V. Lakshmikantham, S. Leela, F.A. McRae, Improved generalized quasilinearization method, Nonlinear Anal. 24 (1995) 1627–1637.

[11] V. Lakshmikantham, Further improvement of generalized quasilinearization, Nonlinear Anal. 27 (1996) 223–227.

[12] V. Lakshmikantham and A.S. Vatsala, Generalized Quasilinearization for Nonlinear Problems, Kluwer Academic Publishers, Netherlands, 1998.

[13] V. Lakshmikantham, N. Shahzad, J.J. Nieto, Methods of generalized quasilinearization for periodic boundary value problems, Nonlinear Anal. 27 (1996) 143–151.

[14] Z. Dricia, F.A. McRae, Devi J. Vasundhara, Quasilinearization for functional differential equations with retardation and anticipation, Nonlinear Anal. 70 (2009) 1763–1775.

[15] R. Bellman, Methods of Nonlinear Analysis, Vol. II, Academic Press, New York, 1973.

[16] L.M. Delves, J.L. Mahamed, Computational Methods for Intrgral Equations, Combridge University Press, (1985) 330-341.

[17] H. Brunner, Collocation methods for volterra integral and related functional equations, Cambridge University Press, 2004.

[18] Datta B., Numerical linear algebra and applications, Brooks/Cole Publishing Company, 1995. p 32.

Finite State Machine Minimization and Row Equivalence Application

Hassan Farhat
University of Nebraska at Omaha

Abstarct- *Finite state machines minimization (finite automata) is a well known problem in formal languages and computer design. For a given automata, effective procedures exist that converts the automaton to a unique equivalent automaton but with possibly fewer number of states.*

While several minimization procedures are devised, we show that some of the procedures do not yield the minimum automata. In particular, we show that, contrary to assumptions, the row equivalence method of minimization does not always yield the unique minimum. We explore the three common methods of minimizations. We then look at the need to employ minimization procedures other than row equivalence to achieve a unique minimum automaton.

1 Introduction

Finite automaton is fundamental in the study of the fields of computer science and computer engineering. It is found in the study of software algorithms as in the case of compilers [2, 3]. And it is important in the design of hardware units in computer engineering [1, 7].

In the study of formal languages, the languages recognized by finite automata are called regular languages. The languages are closed under the set operations: Kleene closure, union, complement and intersection. In addition, given the finite state machines, FSMs, of two regular languages, the automata for the union, complement, and intersection can be easily constructed (an effective procedure exists for the construction) [2]. Effective procedures exist, as well, that reduce the number of state (nodes) in an FSM. For a given FSM there exist a unique equivalent FSM with minimum number of states [2, 3].

In hardware realization, FSMs are mapped to Mealy or Moore machines [7]. In a Mealy machine, the output of the machine is associated with an input and a node (state) while in a Moore machine, the output is associated with a node only. The two machines are equivalent; a procedure exists that maps one machine to the other. In [8] we showed that a minimum automaton does not always yield the optimal hardware design and discussed automata partition [9].

While several procedures are proposed that yield minimum automata [4,5,6], we show that some do not always yield the minimum automaton. In particular we show the row method of minimization as presented in [4, 6] does not yield always a minimum. In [4], the row equivalence algorithm is expanded to include self-loops as a method of minimization. We show that even with this expansion, for some examples, the method does not yield minimum automata.

The paper is organized as follows. In section 2, we present two common methods of automata minimization. In section 3 we look at the row method of minimization. Section 4 presents a counter example and shows that other methods of minimization yield a smaller automaton. The conclusion is given in section 5.

2 Regular Languages and Corresponding Finite State Machines

The definitions of regular languages are found in [2,3][1]. The definition is based on the construction of regular expressions. The regular expression generates (describes) set of elements (words).

A finite automata, also called finite-state machine, M, is a recognizer for a regular language; i.e., for a given word, M will determine if the word is in the regular language. Formally, M, is defined as a quintuple M = (Q, Σ, q0, δ, A) where Q is a finite set of symbols (states), q0 is a special element of Q called the start state, A is a subset of Q (accepting states), Σ is a finite set of symbols (the alphabet), and δ is a function from Q × Σ to Q (the next state function).

The finite-state machine can be described as a directed graph or in tabular form. Example: consider the language over the alphabet {0, 1}. A word in the alphabet is any binary string of 0s and 1s. The language, L, where each word ends in two 0s is a regular language. Figure 1 gives the recognizer finite automat (automata), M, as a directed graph.

[1] This section is presented in [8]

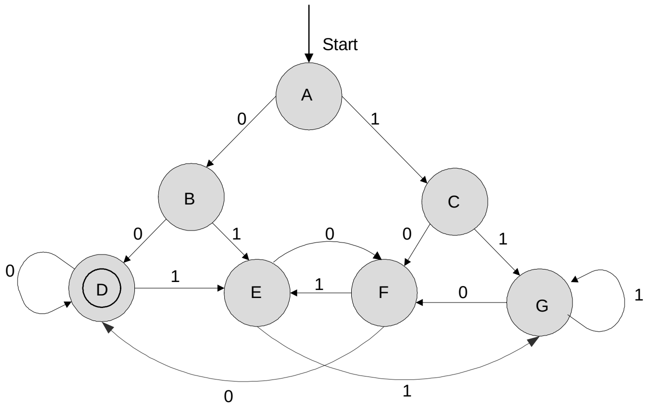


Figure 1: accept states have double circles, start state is labeled with start arrow

In hardware design, we describe the automata a little different. The automata can be described as a Mealy or Moore machine. The accepting states are removed from the definition of the automata. Instead, the machine is presented as a hardware block with inputs and outputs with states assigned binary codes and represented as memory elements. The conversion to Mealy or Moore is simple. Figure 2 shows the Moore conversion. Accepting states have an output of 1 while non-accepting states have an output of 0. For a given input (processed completely), if the final output is 1, the input is a word in the language. An output of zero means the input word is not the language.

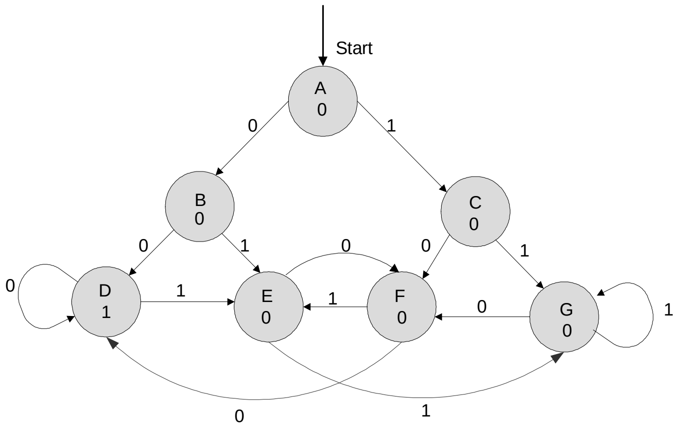


Figure 2: Moore realization, with each state we assign an output (bottom label) ; state D is accepting state (output 1 when in state D)

There are two main minimization procedures that produce the minimized automata: a) the partitioning method, and (b) the implication method [2]. We apply these to our example. The modified tabular representation is shown in Figure 3.

PS	X = 0	X = 1	X = 0, 1
A	B	C	0
B	D	E	0
C	F	G	0
D	D	E	1
E	F	G	0
F	D	E	0
G	F	G	0

Figure 3: Moore tabular form of automata, it includes the output column corresponding to present state

The partition method: Iteratively, place states in groups (partitions) based on the output response from a given state, call the partitions Pi, i corresponds to the iteration number. The 0 iteration contains all the automata states in a single group. P1 places the states in two groups based on the output; those with output zero and those with output 1. P1 : (A, B, C, E, F, G)(D).

Starting with P1 we form new groups based on next states on input 0 and input 1, call these 0-successor and 1 successor respectively. 0-successor(A, B, C, E, F, G) = (B, D, F, D, F). This part of the partition is split if the successor states belong to different previous groups. Split B and F since the next state is D; D and other successor states are in different previous groups. New P2 partition is P2: (A, C, E, G)(B, F)(D). To complete P2 we also form the 1-successors; 1-successor(A, B, C, E, F, G) = (C, E, G, G, E, G); no new groups formed.

Iteratively, we repeat the above to generate P3. The stopping criteria is a point where P(i+1) = Pi. When P3 is formed we find P2 = P3. The modified minimum automata contain three states (A, C, E, G), (B, F) and D.

The implication method of minimization: This procedure applies to incompletely specified automata as well (an automata where the transition function, δ, is not completely defined over the alphabet domain). It is based on a table construction as follows. The table contains columns for every state in the table except the last state; it also contains rows for every state except the first. For a column with label Sj and row Si, the corresponding entry is either X or two pairs of states. The entry is X if the outputs of Si and Sj are different. The table entries are the pairs (Snj0, Sni0) and (Snj1, Sni1) (next state of states Si and Sj on input 0 and 1 respectively). When this is applied to the original automata we obtain the table shown in Figure 4(a).

Pair of states with Xs are not equivalent. From this table we form additional tables by adding new Xs. The stopping criteria: when the new table generated is the same as the previous table. To add Xs, we examine each pair for states entries. We place an X if the entry contains a pair with an X entry already generated in the table.

Lets look at the above example, the first entry checked corresponds to states (A, B) column A and row B. This entry contains the pair (B, D) and (C, E). Since for the entry (B, D) in the current table, Figure 4(a), an X is present, in the new table construction (Figure 4(b)), we place an X in entry (A, B).

	A	B	C	D	E	F
B	B, D C, E					
C	B, F C, G	D, F E, G				
D	X	X	X			
E	B, F C, G	D, F E, G	F, F G, G	X		
F	B, D C, E	D, D E, E	F, D G, E	X	F, D G, E	
G	B, F C, G	D, F E, G	F, F G, G	X	F, F G, G	D, F E, G

(a)

	A	B	C	D	E	F
B	X					
C	B, F C, G	X				
D	X	X	X			
E	B, F C, G	X		X		
F	X		X	X	X	
G	B, F C, G	X		X		X

(b)

Figure 4: Implication method tables

Similar reasoning applies to other entries to obtain Figure 4(b). In the figure pairs of the form (x, x) are removed since it is not possible to replace them with Xs.

On starting with Figure 4(b) and repeating the iteration process, the new table generated is the same as Figure 4(b). Hence, the process stops.

The remaining entries with no Xs represent states that are equivalent. Hence states (A, C), (A, E) (A, G) (B, F) are equivalent. The automata minimization forms an equivalence relation, hence states A, C, E, F and G are equivalent. The final partition is (A, C, E, F, G)(B, F)(D). This is the same grouping obtained using the partition procedure above.

3 The row equivalence method of minimization

The row equivalence method of minimization relies on identification of equivalent rows in a state table as given in Figure 3. It is based on the following definition [6].

Definition: In a state table, two state are said to be equivalent if: a) the next states rows are identical, and b) if the output is the same from each state on each input combination.

In [4], the definition for part (a) was modified so as to allow for cycles in the state diagram. For two states A and B, with identical outputs on all input combinations starting in either state, the two states are equivalent if either or both properties 1 and 2 are satisfied. Property 1: the successor of state A on input x is A, and the successor of state B on input x is B (each state has an edge that loops back to itself on input x). Property 2: If on some input x the successor of state A is state B and the successor of state B is state A.

On repeated identification of equivalent states and reducing the table by keeping only one state from each class of equivalent states, the minimum can be obtained. We apply the method to the example state diagram above. From Figure 3, we note that state B and F are equivalent and states C, E and G are equivalent. On removing states F, E and G we obtain Figure 5.

	NS		OUTPUT
PS	X=0	X=1	X=0, 1
A	B	C	0
B	B	E	0
C	B	C	0
E	B	C	1

(A, C) Equivalent

Figure 5

Note that the states of the table are renamed in cases that reference states that are removed (example row with present state C).

By inspection of the table, we see that states A and C are equivalent. By removing state C we obtain the reduced table found in Figure 6.

	NS		OUTPUT
PS	X=0	X=1	X=0, 1
A	B	A	0
B	B	E	0
E	B	A	1

Rename E as C

Figure 6

On renaming state E, we obtain the minimum automaton, as was done using either method of minimization in the previous section.

We next show that even though the row equivalence method works for the previous example, it does not always yield a minimum. 5

4 A Counter Example

We consider the state table shown in Figure 7. By inspection of the table, there are no identical rows. We then look at two rows that satisfy properties 1 or 2 as stated in section 5. On input x = 0, rows D, E and F have loops (next state = present state). However, the next states are different on input x. Hence property 1 fails. Note that state N is not considered since the output from state N is 1.

	NS		OUTPUT
PS	X = 0	X = 1	X = 0, 1
A	F	B	0
B	E	C	0
C	D	A	0
D	D	M	0
E	E	L	0
F	F	K	0
K	N	C	0
L	N	B	0
M	N	A	0
N	N	N	1

Figure 7

We then check to see if property 2 holds true. For this property to hold true, we need to have two rows i and j (present states i and j) such that on some input x the next state from i is state j and the next state from j is state i. In addition, the remaining elements of the rows should be identical. By inspection, we find this property does not hold true, as well.

Based on the row equivalence method then, we conclude that the above table is already in reduced form. We next show this is not the case by forming a smaller automaton based on the partition method of minimization.

We use the partition method to show the minimum automaton yields fewer states than concluded in the row equivalence method. The 0 iteration contains all the automata states in a single group, P0 = (A, B, C, D, E, F, K, L, M, N). P1 places the states in two groups based on the output; those with output zero and those with output one; P1 = (A, B, C, D, E, F, K, L, M) (N).

From the P1 partition we look at 0-successor(A, B, C, D, E, F, K, L, M) = (F, E, D, D, E, F, N, N, N). Hence the new partition due to 0-successor is (A, B, C, D, E) (K, L, M) (N). The 1-successor(A, B, C, D, E, F, K, L, M) results in (B, C, A, M, L, K, C, B, A) which results in further partitions, P2 = (A, B, C) (D, E, F) (K, L, M) (N). From P2, when we consider the 0-partitions of (A, B, C), (D, E, F), and (K, L, M), we respectively obtain, (F, E, D), (D, E, F) and (N, N, N); no new partitions are generated. Similarly we consider the 1-partitions of (A, B, C), (D, E, F), and (K, L, M). We obtain (B, C, A), (M, L, K) and (C, B, A); no new partitions are generated. Since P3 = P2, the partition in P2 forms the minimum automaton with 4 states only.

5 Conclusion

In this paper we have considered a well known problem in formal languages; the problem of state minimization. While three algorithms are normally given to form the minimum automaton, we showed the row equivalence method does generally result in a minimum automaton. The row equivalence method can be chosen as a preprocessing step in the minimization. However, to guarantee minimum automata it needs to be appended by the implication method of minimization, or the partition method based on the state successor method.

References

[1] N. Weste and D. Harris. "CMOS VLSI design 3rd edition ", Addison Wesley, 2004.

[2] J. Martin. "Introduction to Languages and the Theory of Computation ". McGraw Hill, 1991.

[3] W. Barett, R. Bates, D. Gustafson, J. Couch. "Compiler Construction Theory and Practice, 2nd edition ". SRA publishing, 1979.

[4] V. Nelson, H. Nagle, B. Carroll, J. Irwin. "Digital logic Circuit Analysis and Design ". Prentice Hall, 1995.

[5] R. Katz and B. Gaetano. "Contemporary Logic Design, 2nd edition ". Prentice Hall, 2005.

[6] M. Mano and C. Kime. "Logic and Computer Design Fundamentals, 3rd Edition". Prentice Hall, 2003.

[7] D. Gajski. "Principle of Digital Design". Prentice Hall, 1997.

[8] H. Farhat. "Minimization of State Diagrams and Realization at the Hardware Level". ICIAS, 2010.

[9] L. Benini, G. De Micheli, F. Vemzeulen. "Finite-State Machine Partitioning For Low Power", ISCAS, 1998.

Numerical solution of the Fredholm-Volterra integro-differential equations by the Shannon wavelets

K. Maleknejad[1], **M. Attary**[2]
[1]Department of Mathematics, Fandanesh Institute of Higher Education (FDI), Saveh, 39157-14774, Iran
[2]Department of Mathematics, Karaj Branch, Islamic Azad University, Karaj, Iran

Abstract— *This paper is concerned with obtaining the approximate solution of Fredholm-Volterra integro-differential equations. Properties of the Shannon wavelets and connection coefficients are first presented. We design a numerical scheme for these equations using the Galerkin method incorporated with the Shannon wavelets approximation and the connection coefficients. We will show that using this technique, the Fredholm-Volterra integro-differential equation is transformed to an infinite algebraic system, which can be solved by fixing a finite scale of approximation. The error analysis of the method is also investigated and the reliability and efficiency of the proposed scheme are demonstrated by some numerical experiments.*

Keywords: Fredholm-Volterra integro-differential equations, Shannon wavelets, Connection coefficients, Numerical treatments.

1. Introduction

We consider the following general linear Fredholm-Volterra integro-differential equation of the form:

$$\sum_{n=0}^{m} P_n(t) f^{(n)}(t) = g(t) + \lambda_1 \int_a^b \sum_{i=0}^{p} F_i(t,s) f^{(i)}(s) ds$$
$$+\lambda_2 \int_a^t \sum_{r=0}^{q} K_r(t,s) f^{(r)}(s) ds, \qquad (1)$$

under the mixed conditions

$$\sum_{n=0}^{m-1} \sum_{r'=0}^{d} c_{i'r'}^{n} f^{(n)}(c_{r'}) = \mu_{i'}, \;\; i' = 0,1,...,m-1, \;\; a \le c_{r'} \le b,$$

where the functions $g(t)$, $P_n(t)$, $F_i(t,s)$ $(0 \le i \le p)$ and $K_r(t,s)$ $(0 \le r \le q)$ are defined on the interval $a \le t, s \le b$, $f(t)$ is an unknown function and $c_{i'r'}^{n}$, $c_{r'}$, λ_1, λ_2, $\mu_{i'}$ are constants.

There has been much work on analyzing and numerical methods for the Fredholm-Volterra integro-differential equations. Existence and uniqueness results for (1) have been investigated by many authors under a variety of hypotheses by using different techniques [1-3]. Maleknejad et al. [4] have been concerned with the Taylor polynomials of certain nonlinear Fredholm-Volterra integro-differential equations with algebraic nonlinearity. Akyuz [5] solved (1) by using the Chebyshev polynomials method. In [6], the Taylor polynomials are developed to approximate solution of (1). Recently, Shahmorad [7] has been concerned with the numerical solution of (1) via the Tau method with the arbitrary bases especially Chebychev and Legendre bases.

On the other hand, the Shannon approximation methods have been used widely in numerical solution of many operator equations. These methods have increasingly been recognized as powerful tools for attacking problems in applied physics and engineering [8]. The books [8-9] provide excellent overviews of methods based on Sinc functions for solving ordinary and partial differential equations and integral equations.

In the present work, we propose a method based on the connection coefficients of Shannon wavelets of the solution of linear Fredholm-Volterra integro-differential equations. Our discussion based on the connection coefficients of the Shannon wavelets which proposed by Cattani in [10]. Detailed description and analysis of this technique may be found in [10-14] and references therein. The organization of this paper is as follows: Section 2 outlines some of the main properties and basic definitions of the Shannon wavelets. We clarify how the Shannon wavelets based on the connection coefficients and Galerkin method transform equation (1) to an explicit system of linear algebraic equation in Section 3. The error analysis of the proposed method is given in Section 4. Finally, numerical results are reported to clarify the efficiency of the method.

2. Preliminaries

We first introduce some basic properties of the Shannon wavelets family. The starting point for the definition of these family is the Sinc or Shannon scaling function. The Sinc function is defined on the whole real line by:

$$Sinc(t) = \begin{cases} \frac{sin(\pi t)}{\pi t}, & t \neq 0, \\ 1, & t = 0. \end{cases}$$

The Shannon scaling functions and mother wavelets can be defined as:

$$\begin{cases} \varphi_{j,k}(t) = 2^{j/2} Sinc(2^j t - k) = 2^{j/2} \frac{sin\pi(2^j t - k)}{\pi(2^j t - k)}, & j,k \in Z, \\ \psi_{j,k}(t) = 2^{j/2} \frac{sin\pi(2^j t - k - \frac{1}{2}) - sin 2\pi(2^j t - k - \frac{1}{2})}{\pi(2^j t - k - \frac{1}{2})}, & j,k \in Z, \end{cases}$$

where the Fourier transforms of these functions are respectively obtained from the following equations:

$$\begin{cases} \widehat{\varphi}_{j,k}(\omega) = \frac{2^{-j/2}}{2\pi} e^{-i\omega k/2^j} \chi(\frac{\omega}{2^j} + 3\pi), \\ \widehat{\psi}_{j,k}(\omega) = -\frac{2^{-j/2}}{2\pi} e^{-i\omega(k+1/2)/2^j} [\chi(\frac{\omega}{2^{j-1}}) + \chi(-\frac{\omega}{2^{j-1}}), \end{cases}$$

for $j, k \in Z$,and the characteristic function $\chi(\omega)$ is defined as:

$$\chi(\omega) = \begin{cases} 1, & 2\pi \leq \omega < 4\pi, \\ 0, & otherwise. \end{cases}$$

Using the definition of the inner product of two functions and the Parseval equality, it can be shown easily that the following results hold [10]:

$$\begin{aligned} \Lambda_{kh}^{(l)j} &\equiv < \frac{d^l}{dt^l}\varphi_{o,k}(t), \psi_{j,h}(t) >, \\ \lambda_{kh}^{(l)} &\equiv < \frac{d^l}{dt^l}\varphi_{o,k}(t), \varphi_{o,h}(t) >, \\ \gamma_{kh}^{(l)jj} &\equiv < \frac{d^l}{dt^l}\psi_{j,k}(t), \psi_{j,h}(t) >, \\ \xi_{kh}^{(l)j} &\equiv < \frac{d^l}{dt^l}\psi_{j,k}(t), \varphi_{o,h}(t) >, \end{aligned} \tag{2}$$

where $\delta_{ji}(\delta_{hk})$ denotes the Kronecker delta.

3. The numerical analysis of the method

In order to compute the numerical solution of Fredholm-Volterra integro-differential equation (1), we assume that $f(t)$ be a class of functions such that the following integrals exist and finite:

$$\alpha_k = < f(t), \varphi_{0,k}(t) > = \int_{-\infty}^{\infty} f(t)\varphi_{0,k}(t)dt, \tag{3}$$

$$\beta_{j,k} = < f(t), \psi_{j,k}(t) > = \int_{-\infty}^{\infty} f(t)\psi_{j,k}(t)dt. \tag{4}$$

Due to Theorem 3.2 of [10] and orthogonality conditions (2), the function $f(t)$ and its derivatives can be computed in terms of the Shannon wavelets as:

$$f(t) \cong \sum_{k=-M}^{M} \alpha_k \varphi_{0,k}(t) + \sum_{j=0}^{N} \sum_{k=-M}^{M} \beta_{j,k}\psi_{j,k}(t), \tag{5}$$

$$\frac{d^l}{dt^l} f(t) \cong \sum_{k=-M}^{M} \alpha_k \frac{d^l}{dt^l}\varphi_{0,k}(t) + \sum_{j=0}^{N} \sum_{k=-M}^{M} \beta_{j,k}\frac{d^l}{dt^l}\psi_{j,k}(t). \tag{6}$$

Following [10], we can compute the wavelet decomposition of the functions $\frac{d^l}{dt^l}\varphi_{0,k}(t)$ and $\frac{d^l}{dt^l}\psi_{j,k}(t)$ as follows:

$$\begin{cases} \frac{d^l}{dt^l}\varphi_{0,k}(t) = \sum_{h=-M}^{M} \lambda_{kh}^{(l)}\varphi_{0,h}(t) \\ \qquad + \sum_{j=0}^{N}\sum_{h=-M}^{M} \Lambda_{kh}^{(l)j}\psi_{j,h}(t), \\ \frac{d^l}{dt^l}\psi_{j,k}(t) = \sum_{h=-M}^{M} \xi_{kh}^{(l)j}\varphi_{0,h}(t) \\ \qquad + \sum_{j=0}^{N}\sum_{h=-M}^{M} \gamma_{kh}^{(l)jj}\psi_{j,h}(t), \end{cases} \tag{7}$$

with

$$\Lambda_{kh}^{(l)j} \equiv < \frac{d^l}{dt^l}\varphi_{o,k}(t), \psi_{j,h}(t) >, \tag{8}$$

$$\lambda_{kh}^{(l)} \equiv < \frac{d^l}{dt^l}\varphi_{o,k}(t), \varphi_{o,h}(t) >, \tag{9}$$

$$\gamma_{kh}^{(l)jj} \equiv < \frac{d^l}{dt^l}\psi_{j,k}(t), \psi_{j,h}(t) >, \tag{10}$$

$$\xi_{kh}^{(l)j} \equiv < \frac{d^l}{dt^l}\psi_{j,k}(t), \varphi_{o,h}(t) >, \tag{11}$$

in which they are known as the connection coefficients.

Here, we recall the following relations for computing these coefficients from [10]:

$$\lambda_{kh}^{(l)} = \begin{cases} (-1)^{k-h}\frac{i^l}{2\pi}\sum_{s=1}^{l}\frac{l!\pi^s}{s![i(k-h)]^{l-s+1}}[(-1)^s - 1], & k \neq h, \\ \frac{i^l\pi^{l+1}}{2\pi(l+1)}[1+(-1)^l], & k = h, \end{cases} \tag{12}$$

$$\begin{aligned} \gamma_{kh}^{(l)jj} &= i^l(1-|\mu(h-k)|)\frac{\pi^l 2^{jl-1}}{l+1}(2^{l+1}-1)(1+(-1)^l) \\ &+\mu(h-k)\sum_{s=1}^{l+1}(-1)^{[1+\mu(h-k)](2l-s+1)/2}\frac{l! i^{l-s}\pi^{l-s}}{(l-s+1)!|h-k|^s} \\ &(-1)^{-s-2(h+k)}\times 2^{jl-s-1}\times 2^{l+1}[(-1)^{4h+s} + (-1)^{4k+l}] \\ &-2^s[(-1)^{3k+h+l} + (-1)^{3h+k+s}]. \end{aligned} \tag{13}$$

$$\Lambda_{kh}^{(l)j} = \xi_{kh}^{(l)j} = 0. \tag{14}$$

Based on these relations, (6) can be rewritten as:

$$\begin{aligned} \frac{d^l}{dt^l}f(t) &\cong \sum_{k=-M}^{M}\alpha_k\sum_{h=-M}^{M}\lambda_{kh}^{(l)}\varphi_{0,h}(t) \\ &+ \sum_{j=0}^{N}\sum_{k=-M}^{M}\beta_{j,k}\sum_{h=-M}^{M}\gamma_{kh}^{(l)jj}\psi_{j,h}(t). \end{aligned} \tag{15}$$

In order to obtaining the numerical solution, we compute functions $\frac{d^n}{dt^n}f(t)$, $\frac{d^i}{dt^i}f(t)$ and $\frac{d^r}{dt^r}f(t)$ from (15) and substituting these relations in equation (1), so we can define the residual vector as follows:

$$\begin{aligned} r_{(2M+1)(N+2)} = \\ -g(t) + [I_{1k}(t) + \lambda_1 I_{2k}(t) + \lambda_2 I_{3k}(t)]\sum_{k=-M}^{M}\alpha_k \\ +[J_{1jk}(t) + \lambda_1 J_{2jk}(t) + \lambda_2 J_{3jk}(t)]\sum_{j=0}^{N}\sum_{k=-M}^{M}\beta_{j,k}, \end{aligned}$$

where

$$I_{1k}(t)=\sum_{n=0}^{m}P_n(t)\sum_{h=-M}^{M}\lambda_{kh}^{(n)}\varphi_{0,h}(t),$$

$$I_{2k}(t)=-\int_a^b\sum_{i=0}^{p}F_i(t,s)\sum_{h=-M}^{M}\lambda_{kh}^{(i)}\varphi_{0,h}(s)ds,$$

$$I_{3k}(t)=-\int_a^t\sum_{r=0}^{q}K_r(t,s)\sum_{h=-M}^{M}\lambda_{kh}^{(r)}\varphi_{0,h}(s)ds,$$

$$J_{1jk}(t)=\sum_{n=0}^{m}P_n(t)\sum_{h=-M}^{M}\gamma_{kh}^{(n)jj}\psi_{j,h}(t),$$

$$J_{2jk}(t)=-\int_a^b\sum_{i=0}^{p}F_i(t,s)\sum_{h=-M}^{M}\gamma_{kh}^{(i)jj}\psi_{j,h}(s)ds,$$

$$J_{3jk}(t)=-\int_a^t\sum_{r=0}^{q}K_r(t,s)\sum_{h=-M}^{M}\gamma_{kh}^{(r)jj}\psi_{j,h}(s)ds.$$

Also, the mixed conditions give m equations as follows:

$$[\sum_{n=0}^{m-1}\sum_{r'=0}^{d}c_{i'r'}^{n}\sum_{h=-M}^{M}\lambda_{kh}^{(n)}\varphi_{0,h}(t)]\sum_{k=-M}^{M}\alpha_k$$

$$+[\sum_{n=0}^{m-1}\sum_{r'=0}^{d}c_{i'r'}^{n}\sum_{h=-M}^{M}\gamma_{kh}^{(n)jj}\psi_{j,h}(t)]\sum_{j=0}^{N}\sum_{k=-M}^{M}\beta_{j,k}=\mu_{i'}. \tag{16}$$

Let us introduce ϕ and c, for $j = 0, ..., N$ and $k = -M, ..., M$, by

$$\begin{cases}\phi_{k+M+1}:=\varphi_{0,k},\\ \phi_{(2M+1)(j+1)+k+M+1}:=\psi_{j,k},\\ c_{k+M+1}:=\alpha_k,\\ c_{(2M+1)(j+1)+k+M+1}:=\beta_{j,k}.\end{cases} \tag{17}$$

With this notation, we have

$$\sum_{k=-M}^{M}\alpha_k\varphi_{0,k}(t)+\sum_{j=0}^{N}\sum_{k=-M}^{M}\beta_{j,k}\psi_{j,k}(t)=\sum_{i=1}^{(2M+1)(N+2)}c_i\phi_i(t). \tag{18}$$

Due to Galerkin method, we can compute c_i, $i = 1, ..., (2M + 1)(N + 2)$, such that

$$< r_{(2M+1)(N+2)},\phi_{j'}>=0,\quad j'=1,...,(2M+1)(N+2)-m. \tag{19}$$

With respect to the relations (16) and (19), we have $(2M + 1)(N + 2)$ algebraic equations. Finally, we end up with system of equations whose solution gives the unknown coefficients α_k and $\beta_{j,k}$. Therefore the desired approximation to the solution of (1) can be obtained from (5).

4. Error analysis

In this section, we discuss the convergence and error estimation of the Shannon wavelets approximation for the linear Fredholm-Volterra integro-differential equation (1). In our convergence analysis, firstly we recall the following auxiliary theorem from [15]. Throughout this section, $\varphi_{0,k}$ and $\psi_{j,k}$ are defined as the Shannon scaling functions and mother wavelets.

Theorem 4.1. Let $\{\psi_{j,k}(t)\}_{j,k\in Z}$ be a sequence of wavelets in the Shannon wavelets family as described in Section 2. Assume that $f^{(l)}(t)\in L_2(R)$. Then series

$$\sum_{j=-\infty}^{N-1}\sum_{k=-\infty}^{\infty}\langle f(t),\psi_{j,k}(t)\rangle\psi_{j,k}^{(l)}(t),$$

converges to $f^{(l)}(t)$, for each $l = 0, 1, 2, ...$

In order to examine the convergence of the approximate solution $\widetilde{f}(t)$ to the exact solution $f(t)$ of equation (1), notice that

$$\widetilde{f}(t)=\sum_{k=-\infty}^{\infty}\alpha_k\varphi_{0,k}(t)+\sum_{j=0}^{N-1}\sum_{k=-\infty}^{\infty}\beta_{j,k}\psi_{j,k}(t).$$

We are now ready to prove the following theorem, which is as a consequence of Theorem 4.1.

Theorem 4.2. Let $f(t)$ and $\widetilde{f}(t)$ be the exact and approximate solutions of the equation (1), respectively. Moreover, suppose α_k and $\beta_{j,k}$ are given by (3) and (4). If $f^{(l)}(t)\in L_2(R)$, then the obtained approximation solution of the proposed method for equation (1) converges to the exact solution.

Proof. Note that

$$\begin{aligned}\widetilde{f}(t)\quad &=\textstyle\sum_{k=-\infty}^{\infty}\langle f(t),\varphi_{0,k}(t)\rangle\varphi_{0,k}(t)\\ &+\textstyle\sum_{j=0}^{N-1}\sum_{k=-\infty}^{\infty}\langle f(t),\psi_{j,k}(t)\rangle\psi_{j,k}(t)\\ &=\textstyle\sum_{j=-\infty}^{N-1}\sum_{k=-\infty}^{\infty}\langle f(t),\psi_{j,k}(t)\rangle\psi_{j,k}(t).\end{aligned} \tag{20}$$

Due to the Theorem 4.1, we have

$$\|D^{(l)}[\sum_{j=-\infty}^{N-1}\sum_{k=-\infty}^{\infty}\langle f(t),\psi_{j,k}(t)\rangle\psi_{j,k}(t)-f(t)]\|_2\to 0,$$

$$as\quad N\to\infty. \tag{21}$$

or equivalently

$$\|\sum_{j=-\infty}^{N-1}\sum_{k=-\infty}^{\infty}\langle f(t),\psi_{j,k}(t)\rangle\psi_{j,k}^{(l)}(t)-f^{(l)}(t)\|_2\to 0,$$

$$as\quad N\to\infty. \tag{22}$$

Based on equation (20), relation (21) can be rewritten as

$$\|D^{(l)}[\sum_{k=-\infty}^{\infty}\langle f(t),\varphi_{0,k}(t)\rangle\varphi_{0,k}(t)$$
$$+\sum_{j=-\infty}^{N-1}\sum_{k=-\infty}^{\infty}\langle f(t),\psi_{j,k}(t)\rangle\psi_{j,k}(t)-f(t)]\|_2\to 0,$$

$as\ \ N\to\infty$,
and so

$$\|\sum_{k=-\infty}^{\infty}\langle f(t),\varphi_{0,k}(t)\rangle\varphi_{0,k}^{(l)}(t)$$
$$+\sum_{j=-\infty}^{N-1}\sum_{k=-\infty}^{\infty}\langle f(t),\psi_{j,k}(t)\rangle\psi_{j,k}^{(l)}(t)-f^{(l)}(t)\|_2\to 0,$$

$as\ \ N\to\infty$.

Substituting relations (3), (4) and (7) in above equation, we get

$$\lim_{N\to\infty}[\sum_{k=-\infty}^{\infty}\alpha_k\sum_{h=-\infty}^{\infty}\lambda_{kh}^{(l)}\varphi_{0,h}(t)$$
$$+\sum_{j=0}^{N-1}\sum_{k=-\infty}^{\infty}\beta_{j,k}\sum_{h=-\infty}^{\infty}\gamma_{kh}^{(l)jj}\psi_{j,h}(t)]=f^{(l)}(t),$$

finally, we can write

$$\lim_{N\to\infty}[\sum_{k=-\infty}^{\infty}\alpha_k\sum_{h=-\infty}^{\infty}\lambda_{kh}^{(n)}\varphi_{0,h}(t)$$
$$+\sum_{j=0}^{N-1}\sum_{k=-\infty}^{\infty}\beta_{j,k}\sum_{h=-\infty}^{\infty}\gamma_{kh}^{(n)jj}\psi_{j,h}(t)]=f^{(n)}(t),$$

$$\lim_{N\to\infty}[\sum_{k=-\infty}^{\infty}\alpha_k\sum_{h=-\infty}^{\infty}\lambda_{kh}^{(i)}\varphi_{0,h}(t)$$
$$+\sum_{j=0}^{N-1}\sum_{k=-\infty}^{\infty}\beta_{j,k}\sum_{h=-\infty}^{\infty}\gamma_{kh}^{(i)jj}\psi_{j,h}(t)]=f^{(i)}(t),$$

$$\lim_{N\to\infty}[\sum_{k=-\infty}^{\infty}\alpha_k\sum_{h=-\infty}^{\infty}\lambda_{kh}^{(r)}\varphi_{0,h}(t)$$
$$+\sum_{j=0}^{N-1}\sum_{k=-\infty}^{\infty}\beta_{j,k}\sum_{h=-\infty}^{\infty}\gamma_{kh}^{(r)jj}\psi_{j,h}(t)]=f^{(r)}(t),$$

and the above relations prove the theorem. □

In order to compute an error bound for the solution of linear Fredholm-Volterra integro-differential equation (1), we determine an upper bound for $|f(t)-\widetilde{f}_M(t)|$ and $|f^{(l)}(t)-\widetilde{f}_M^{(l)}(t)|$. For this purpose, we recall the following theorem from [16].

Theorem 4.3. Let $f(t)$ defined as follows:

$$f(t)\cong\sum_{h=-K}^{K}\alpha_h\varphi_{0,h}(t)+\sum_{j=0}^{N}\sum_{k=-S}^{S}\beta_{j,k}\psi_{j,k}(t), \tag{23}$$

then the error of approximation (23), is given by

$$|f(t)-\sum_{h=-K}^{K}\alpha_h\varphi_{0,h}(t)+\sum_{j=0}^{N}\sum_{k=-S}^{S}\beta_{j,k}\psi_{j,k}(t)|$$
$$\le|f(-K-1)+f(K+1)-\tfrac{3\sqrt{3}}{\pi}[f(2^{-N-1}(-S-\tfrac{1}{2}))$$
$$+f(2^{-N-1}(S+\tfrac{3}{2}))]|.$$

In this position, our main concern is to determine an upper bound for (15). The behavior of $|f^{(l)}(t)-\widetilde{f}_M^{(l)}(t)|$ has been investigate under some assumptions in [16].

Theorem 4.4. Let $f_M^{(l)}(t)$ be the approximate solution of derivatives of equation (1), which is given by (15), then there exist constants c_{1l} and c_{2l} independent of N and M, which M and N refer to the given values of j,k, such that

$$|f^{(l)}(t)-\widetilde{f}_M^{(l)}(t)|$$
$$\le|C_{1l}(f(-M-1)+f(M+1))-C_{2l}[\frac{3\sqrt{3}}{\pi}[f(2^{-N-1}(-M-\frac{1}{2}))$$
$$+f(2^{-N-1}(M+\frac{3}{2}))]]|,$$

where

$$\begin{cases} C_{1l}=Max\{|\sum_k\sum_h\lambda_{kh}^{(l)}|\},\\ C_{2l}=Max\{|\sum_k\sum_h\gamma_{kh}^{(l)jj}|\}.\end{cases} \tag{24}$$

Proof. The error of the approximation (15) can be written as:

$$|f^{(l)}(t)-\widetilde{f}_M^{(l)}(t)|$$
$$=|\frac{d^l}{dt^l}f(t)-\sum_{k=-M}^{M}\alpha_k\sum_{h=-M}^{M}\lambda_{kh}^{(l)}\varphi_{0,h}(t)$$
$$-\sum_{j=0}^{N}\sum_{k=-M}^{M}\beta_{j,k}\sum_{h=-M}^{M}\gamma_{kh}^{(l)jj}\psi_{j,h}(t)|$$
$$=|E_1+E_2|,$$

where

$$E_1=\sum_{k=-\infty}^{-M-1}\alpha_k\sum_{h=-\infty}^{-M-1}\lambda_{kh}^{(l)}\varphi_{0,h}(t)+\sum_{k=M+1}^{\infty}\alpha_k\sum_{h=M+1}^{\infty}\lambda_{kh}^{(l)}\varphi_{0,h}(t),$$

$$E_2=\sum_{j=N+1}^{\infty}[\sum_{k=-\infty}^{-M-1}\beta_{j,k}\sum_{h=-\infty}^{-M-1}\gamma_{kh}^{(l)jj}\psi_{j,h}(t)$$
$$+\sum_{k=M+1}^{\infty}\beta_{j,k}\sum_{h=M+1}^{\infty}\gamma_{kh}^{(l)jj}\psi_{j,h}(t)]|.$$

Note that

$$E_1 \leq max[\sum_{k=-\infty}^{-M-1} \alpha_k \sum_{h=-\infty}^{-M-1} \lambda_{kh}^{(l)} \varphi_{0,h}(t) + \sum_{k=M+1}^{\infty} \alpha_k \sum_{h=M+1}^{\infty} \lambda_{kh}^{(l)} \varphi_{0,h}(t)],$$

$$E_2 \leq max \sum_{j=N+1}^{\infty} [\sum_{k=-\infty}^{-M-1} \beta_{j,k} \sum_{h=-\infty}^{-M-1} \gamma_{kh}^{(l)jj} \psi_{j,h}(t) + \sum_{k=M+1}^{\infty} \beta_{j,k} \sum_{h=M+1}^{\infty} \gamma_{kh}^{(l)jj} \psi_{j,h}(t)],$$

Actually, as stated in [16, p. 12] and [17] the following relations hold

$$max[\sum_{k=-\infty}^{-M-1} \alpha_k \sum_{h=-\infty}^{-M-1} \varphi_{0,h}(t) + \sum_{k=M+1}^{\infty} \alpha_k \sum_{h=M+1}^{\infty} \varphi_{0,h}(t)] \leq f(-M-1) + f(M+1),$$

and

$$max \sum_{j=N+1}^{\infty} [\sum_{k=-\infty}^{-M-1} \beta_{j,k} \sum_{h=-\infty}^{-M-1} \psi_{j,h}(t) + \sum_{k=M+1}^{\infty} \beta_{j,k} \sum_{h=M+1}^{\infty} \psi_{j,h}(t)] \leq \frac{-3\sqrt{3}}{\pi}[f(2^{-N-1}(-M-\frac{1}{2}) + f(2^{-N-1}(M+\frac{3}{2}).$$

Based on these relations, we have the following inequalities

$$E_1 \leq C_{1l}\{f(-M-1)+f(M+1)\}, \tag{25}$$

$$E_2 \leq C_{2l}\{\frac{-3\sqrt{3}}{\pi}[f(2^{-N-1}(-M-\frac{1}{2})+f(2^{-N-1}(M+\frac{3}{2})\}. \tag{26}$$

Finally the assertion follows in consequence of equations (25) and (26). □

5. Numerical results

Here, we consider two test problems from [4, 7] and report the numerical results by the proposed method. The "Maximal Error" refers to the maximal difference between approximation and exact solution.

$$\begin{cases} f^{''}(t) + tf(t) = \int_0^t t^2 e^s f(s)ds - (1+t)cos(t) \\ -\frac{1}{2}(e^t(cos(t)+sin(t))-1)t^2, \\ f(0) = 1, \\ f^{'}(0) = 0, \qquad 0 \leq t \leq 1 \end{cases} \tag{5.1}$$

with the exact solution $f(t) = cos(t)$.

$$\begin{cases} tf^{''}(t) - tf^{'}(t) + 2f(t) = \frac{1}{12}t^4 - \frac{1}{6}t^3 - \frac{1}{2}t^2 - \frac{13}{6}t + \frac{17}{12} \\ + \int_0^1 (t+s)f(s)ds + \int_0^t (t-s))f(s)ds, \\ f^{'}(0) - 2f(1) + 2f(0) = 1, \\ f(0) = 1, \end{cases} \tag{5.2}$$

$f(t) = -t^2 + t + 1.$

Equation (5.1) was solved in [4] by a method based on Taylor polynomial for $N = 5, 6, 7$. In this case, the best result has the error of order $O(10^{-6})$. The comparison of the numerical results between the proposed method and method in [4] are given in Table 1.

Table 1. The errors for (5.1) at $t = 1$

N	M	Present method	Method in [4]
3	2	$.83E-7$	$.27E-3$
4	2	$.58E-9$	$.58E-6$
5	3	$.34E-12$	$.58E-6$
9	3	$.28E-15$	$-$

For computational details and numerical implementation of the proposed algorithm, we take $N = M = 1$ for (5.2), so the following values will be obtained:

$\lambda_{kh}^{(1)}$	$k=-1$	$k=0$	$k=1$
$h=-1$	0	1	$-\frac{1}{2}$
$h=0$	-1	0	1
$h=1$	$\frac{1}{2}$	-1	0

$\gamma_{kh}^{(1)11}$	$k=-1$	$k=0$	$k=1$
$h=-1$	0	-6	-1
$h=0$	6	0	-6
$h=1$	1	6	0

From (16) and (19), we obtain the following results:

$$\begin{cases} \alpha_{-1} = -0.552305, \alpha_0 = 0.412517, \alpha_1 = 0.992934, \\ \beta_{0,-1} = -0.202466, \beta_{0,0} = -0.16766, \beta_{0,1} = -0.8964427, \\ \beta_{1,-1} = 0.761246, \beta_{1,0} = 0.125023, \beta_{1,1} = 0.59325, \end{cases}$$

Substituting these values in equation (5), an approximate solution will be obtained. The numerical results are given in Table 2.

Table 2. Numerical results of example (5.2) for $N = M = 1$

t	Approximate solution	Exact solution
0	1.042000	1.0000
0.2	1.161625	1.1600
0.4	1.252500	1.2400
0.6	1.234375	1.2400
0.8	1.194375	1.1600
1	0.987000	1.0000

To investigate the high accurate solution of the present method, the computational results of (5.2) have been reported in Tables 3.

Table 3. Numerical results for (5.2) at $t = 1$

N	M	Maximal Error
2	2	$2.16E-7$
3	2	$2.58E-8$
7	4	$3.37E-12$
8	4	$1.56E-15$
9	5	$5.16E-17$

6. Conclusions

In this research, the Shannon wavelets approximation including the connection coefficients was proposed for the Fredholm -Volterra integro-differential equation. We observe here significant improvements have been obtained compared with the numerical results reported by others. We can improve the accuracy of the solution by selecting the large values of M, N.

References

[1] B. G. Pachpatte, "On higher order Volterra-Fredholm integro-differential equation," *Fasciculi Mathematici*, vol. 37, 2007.

[2] R. P. Agarwal, "Boundary value problems for high ordinary differential equations," *World Scientific*, Singapore, 1986.

[3] V. Muresan, "Existence, Uniqueness and data dependence for the solutions of some integro-differential equations of mixed type in Banach space," *J. Analysis and its Application*, vol. 23, PP. 205–216 2004.

[4] K. Maleknejad, Y. Mahmoudi, "Taylor polynomial solution of high-order nonlinear Volterra-Fredholm integro-differential equations," *Appl. Math. Comput.*, vol. 145, PP. 641–653, 2003.

[5] A. Akyuz-Dasicioglu, "A Chebyshev polynomial approach for linear Fredholm-Volterra integro-differential equations in the most general form," *Appl. Math. Comput.* vol. 181, PP. 103–112, 2006.

[6] S. Yalcinbasm, M. Sezer, "The approximate solution of high-order linear Volterra-Fredholm integro-differential equations in terms of Taylor polynomial," *Appl. Math. Comput.* vol. 112, PP. 291–308, 2000.

[7] S. Shahmorad, "Numerical solution of the general form linear Fredholm-Volterra integro-differential equations by the Tau method with an error estimation," *Appl. Math. Comput.* vol. 167, PP. 1418–1429, 2005.

[8] F. Stenger, "Numerical methods based on Sinc and analytioc functions," *Springer-Verlag*, New York, 1993.

[9] J. Lund, K. Bowers, "Sinc methods for quadrqture and differential equations," *SIAM, Philadelphia*, PA, 1992.

[10] C. Cattani, "Shannon wavelets theory," *Math. Probl. Eng.* Article ID 164808, PP. 1-24, 2008.

[11] C. Cattani, "Connection coefficients of Shannon wavelets," *Math. Model. Anal.* vol. 11, PP. 117–132, 2006.

[12] C. Cattani, "The wavelets based technique in the dispersive wave propagation," *Internat. Appl. Mech.* vol. 39, PP. 132–140, 2003.

[13] C. Cattani, "Shannon wavelet analysis," *in Proceeding of the 7th International Conference on Computational Science(ICCS'07)*, Beijing, China, LNCS 4488, Part II, Springer-Verlag Berlin Heidelberg, 2007.

[14] C. Cattani, "Harmonic wavelets towards the solution of nonlinear PDE," *Compute. Math. Appl.* vol. 50, PP. 1191–1210, 2005.

[15] J. Pandey, "Shannon wavelet series expansion of a class of generalized functions," *Journal of wavelet and theory and application,* vol. 1, PP. 65–81, 2007.

[16] C. Cattani, "Shannon wavelets for the solution of Integro-differential equations," *Math. Probl. Eng.* PP. 1–22, 2010.

[17] K. Maleknejad, M. Attary, "An efficient numerical approximation for the linear class of Fredholm integro-differential equations based on Cattani's method," *Commun. Nonlinear. Sci. Numer.*, vol. 16, pp. 2672–2679, 2010.

Study on Oracle Database Security Strategy of User Identification

Wenrong Jiang[1], Anbao Wang[1], Jian Chen[1], Jihong Yan[1]
[1] The school of computer and information, Shanghai Second polytechnic University
201209 Shanghai, China

Abstract - *This paper introduces the way to protect the user identification in Oracle 9i at the network interface layer under the network environment. It analyzes the flaws and bugs that exist in user identification in Oracle 9i, proposes the solution and makes the security policy of user identification. At last it analyzes the database performance in this safe protection mode, shows the prospect of the security strengthening software.*

Keywords: Oracle database, user identification, security policy, password, security strengthening software

1 Introduction

With the computer and network widely used in all areas of society, particularly the growing popularity of Internet networks and network viruses, hackers, spyware continuing to steal, tamper and abuse, security issues become increasingly important. But some Internet safety experts are mostly focused on the security of network services and operating systems, they think that once the key network services are locked and the vulnerabilities of the operating system are amended, all the applications on the server will be able to get secure. However, modern database systems with a variety of features and performance configuration in use are likely to jeopardize the confidentiality, validity and integrity of the data. The back door of these systems and their own safety defects not only seriously endanger the security of the database, but also affect the server's operating system and other information systems. At present, a large database of U.S. production, Oracle, Sybase, DB2, SQL Server are widely used in computer application field. Oracle database system is recognized by the majority of users for its outstanding performance. In IDC's 2004 ranking of the database market, Oracle database firmly occupies the first place with the amount of 41.3% of the market share.

As we all know, each DBMS user first needs to enter a user name, it must also provide a user ID to identify the user's authenticity and reliability. This is the first defense of database security, but also one of the relatively weak links.

The paper proposes the corresponding security policy to the existing security risks and bugs of user identification in Oracle 9i database and protects user identification of the Oracle 9i database on the network interface layer with the purpose of protecting all the application programs on the application level.

2 Related Work

2.1 *Requirements to User Identification in the Level of Computer Security*

User identification mainly refers to the identification of the database user ID. According to the Trusted Computer System Evaluation Criteria (TCSEC) of the U.S. Department of Defense (DOD) and National Bureau of Standards (NISI), it is generally believed that a database management system with B1-level security features is adaptable to the basic needs of the secure online data access. But so far in Chinese domestic DBMS used, part of the desktop database doesn't have security function, the security function of most of the database system only reaches C1 level, such as SQL Server or C2 level, such as Oracle (the U.S. blockade against China on the database security software of B1 level and above), but almost no database with B1-level security function was used in the applications. Our project is, by studying the security strategy of Oracle 9i, to develop the security strengthening software of Oracle 9i in the network interface layer so that the visit to Oracle database through security strengthening software reaches B1 security level. In the B1 security level, the requirements to the user identification are as follows:

2.1.1 Discretionary Access Control

In computer information system definition of trusted computing base and control system, the user access to the named object is named. Implementation mechanisms (such as: Access Control List) allow the named user to fix and control the share of the object with the identity of the user and (or) the user group; prevent unauthorized users to read sensitive information.

2.1.2 Identity Authentication

The initial implementation of trusted computing base of computer information systems first requires the user to identify their own identity and to use protective mechanisms (for example: password) to authenticate the user's identity and prevent unauthorized users from accessing the user authentication data.

2.1.3 Unique and Non-reusable User ID

By providing users with unique identifier, the computer information system trusted computing base enables users to be responsible for their actions. The failure of the user ID is protected by setting the non-reusable user ID.

2.1.4 Using a password or the high strength security mechanisms

Use password or high-strength security mechanisms, such as the IC card information, biometric information and other specific identification information to authenticate the identity, and require users to authenticate each time when logging in. Authentication information should be invisible and protected through data encryption in the process of storage and transmission.

2.2 *Security Risks and Bugs of User Identification when Oracle 9i is Used Domestically*

At present, the domestic Oracle 9i system sets a unique identity and password for each Oracle user. Only legitimate users who use the correct user ID and password can log on and use the Oracle database system. The present authentication mechanisms of Oracle DBMS meet the requirements of C2-level security standards, but fail to reach the basic requirements of current online data access of safety. As to user identification, Oracle 9i has the following bugs:

2.2.1 Security risks of the default user account and password

Oracle 9i is a database management system based on web service applications. During installation, the installation of the components can create as many as 160 different default user account (including sys and system, which database administrators are familiar with), these accounts use the known default passwords, the majority of which are the same as the account number. Among these default accounts, some are authorized to manage and have remote access to the database, an intruder can have full access to the database and damage the system through these accounts..

2.2.2 Bugs of unauthenticated server user accessing sensitive information

The default installation of Oracle 9i includes the apache web services and other apache services. Under the default installation of Oracle 9i, Certified remote users can access sensitive information, including dynamic monitoring services. So all the certified remote users of Oracle 9i (non-server authenticated user) can access the server information and invade the server system.

2.2.3 Bugs of the disclosure of default configuration file information

Oracle 9i includes two important configuration files called "XSQLConfig.xml" and "soapConfig.xml", which contains such sensitive information as the user name and password of the database. These files can be accessed by the unauthenticated remote user, remote users can access and read files through virtual directory

2.3 *Security Configuration Policy of Oracle 9i in Terms of the User Identity*

Domestic Oracle 9i has reached C2 level, the user's security policies it provides in terms of the user identity are as follows:

Each database user has a password, which can be used to prevent unauthorized access. Oracle checks the password, making sure the password meets the following conditions:

- at least four characters
- different from the user name
- not matching the internal simple vocabulary
- having at least three characters different from the previous password (if any)

Each time a password is created or modified, the Oracle can be notified to check whether these features can be met.

In Oracle 9i, the password security policy can also be set by modifying the user profile, password complexity can be customized by oneself. The following parameters are related to the password security:

- maximum continuous failure of the user visits: FAILED_LOGIN_ ATTEMPTS
- time limit of the user password: PASSWORD_LIFE_TIME
- lock time after the password fails: PASSWORD_GRACE_TIME
- lock time when user logs in more than the effective number of times PASSWORD_LOCK_TIME:
- number to keep the password history record: PASSWORD_REUSE _MAX
- time to keep the password history record: PASSWORD_REUSE_TIME
- audit function of the password complexity: PASSWORD_VERIFY _FUNCTION

By default, these security policies don’t work.

3 Safety Design of User Identification

3.1 *Interface Design of the Security Reinforcement*

Oracle 9i is a database management system with completely closed-end interface, but how to implement security features on the basis that Oracle system source code and the copyright can not be accessed is the problem the project aims to solve. Aiming at Oracle 9i user ID security, the article will put forward security design plans and carry them out.

First, consider the database is mainly used for network security, thus the enhanced system security can be set on the Oracle network interface.

Second, Oracle has two network interfaces: one is the C / S mode, enter the Oracle database through ODBC; the other is B / S mode, with the Web Server, access Oracle via ODBC / JDBC.

Based on the above two cases, the Oracle security reinforcement system we have designed and developed should be:

(1) Logically it is a completely independent system with Oracle.

(2) It consists of two parts, the first part is mandatory access control function; second part is the network interface function, which is to build a closed secure path to Oracle accessed by the C/S mode and B/S mode. It requires that the access to Oracle through this interface must be converted to ODBC / JDBC interfaces.

(3) The physical location of this security reinforcement is outside the Oracle, but within the interface of the Internet user access. Its composition diagram is as Figure 1.

Thus, when an application issues SQL statement, before the implementation of the Oracle database, the database security reinforcement device also needs to mandatorily access control checks apart from Oracle's own security checks. Only after the passage of this examination can Oracle database execute SQL statements, and finally the results are returned directly to the application. The reason why the database security reinforcement device is set on the network interface layer is that any access to the database through the data network has to go through the network interface and has to be certified by database security strengthening device, which reaches the effect of mandatory access control checks.

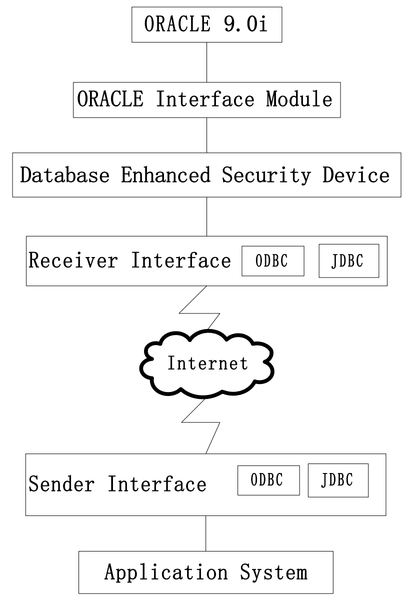


Fig. 1. Enforce to access Oracle database through security strengthening software.

3.2 *Function Design of the User ID of Security Reinforcement*

Next the database security reinforcement device should be designed to meet B1 level security standards. In terms of the user id security, security reinforcement device is to achieve the following functions:

3.2.1 Authority Design

Different types of users with different identities are given different privileges. In security strengthening software, the identity of the general database users and administrators who are responsible for general management of the DAC security must be defined. Only administrators can degrade into general database users, who are not allowed to upgrade into administrators.

3.2.2 Login Authentication Design

What Oracle 9i defaults is password authentication, as password can easily be stolen, so the Oracle 9i supports multiple authentication methods: CyberSafe, Kerberos, DCE, SecurID, Identix Biometric, RADIUS, SSL and PKI. Security reinforcement device not only supports the above authentication methods, but also provides the function of IC card authentication.

3.2.3 Design for Non-reusable User ID and the Valid Date

In the security system, the user ID can not be reused and the user valid date is set, which can better control the valid date of the temporary users, the general database users and the security administrators.

3.2.4 Limit User's Login Time and IP

As there exists security buds in Oracle's TNS listener, which will lead the intruder to get super-user privileges of the operating system or the right to modify data in the database. Therefore, the restrictions to the user's login time everyday and the IP address can make up for this vulnerability.

3.2.5 Function Design for the User Suspension

In order to temporarily block the use of an account, safety reinforcement software provides the account with the function of suspension and reconciliation

3.2.6 Security Policy Settings

In addition that the password policy setting in Oracle 9i is inherited , but also the settings of the maximum number of users creating a session at the same time and the user not using the time limits and other security settings are increased.
By setting these security features, the flaws and bugs of the user ID in Oracle 9i can be made up and the security level to B1 can be reached.

3.2.7 Transmission Encryption

Oracle uses TNS protocol to transfer data, as in the transmission process it can not be guaranteed that the data is intercepted and even modified, so the transmission must be encrypted.

4 Security Implementation

4.1 *The Database Design of the Security Reinforcement User ID part*

The database design of the security reinforcement user ID part aims at authenticating on the user ID in checking mandatory access control. According to the above functions, the database is mainly composed of three tables: table of database users, user login table and security policy table, the table structure and its correlation can be seen in Figure 2.

4.2 *The SQL command design of the user ID security strengthening device*

In addition to the redefinition of the existing SQL commands of Oracle 9i, some security-related SQL commands must also be added according to security requirements. SQL command set of the user ID is listed as follows:

- Create a user: CREATE USER <username> IDENTIFIED BY <password>
- Delete user: DROP USER <username>
- Change password: ALTER USER <username> IDENTIFIED BY <password>

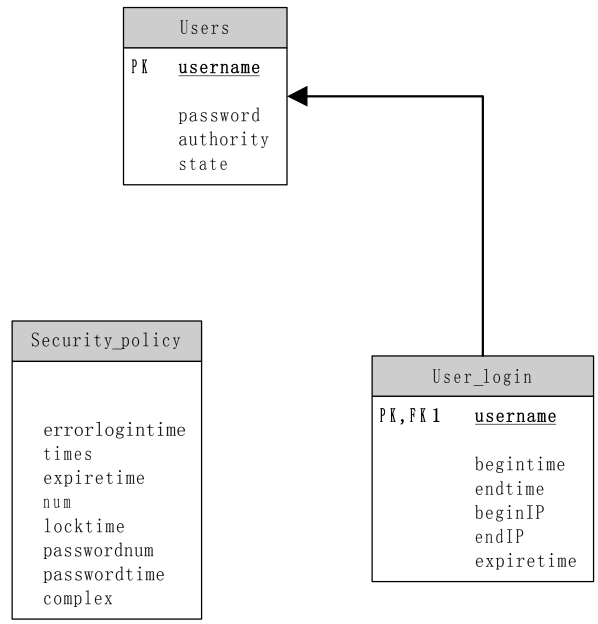


Fig. 2. security database design of the user ID part

- Browse users: SELECT USERNAME FROM USER
- Set pending: ALTERDACSTATE <username> <state>
- Users period set: TCBSETUPEXPIRE DAC <username> <expiretime>
- time limit of the user access: TCBSETPERIOD DAC <username> <begintime> <endtime>
- user IP address restrictions: TCBSETIP DAC <username> <beginIP> <endIP>
- maximum continuous failure of the user visits: TCBSETFAILTIME DAC <times>
- -time limit of the user password: TCBSETPWDDATE DAC <expiretime>
- -lock time after the password fails:: TCBLOCKTIME DAC <locktime>
- : lock time when user logs in more than the effective number of times TCBERRORLOGIN DAC <errorlogintime>
- number to keep the password history record: TCBPASSWORDNUM DAC <passwordnum>
- time to keep the password history record: TCBPASSWORDTIME DAC <passwordtime>
- : -audit function of the password complexity TCBCOMPLEX DAC <complex>
- Users without time limit set: TCBSETUNUSETIME DAC <times>
- User maximum amount of sessions set: TCBSETMAXSESSION DAC <num>

Some commands (such as valid user settings, user settings of the maximum number of sessions) which are not included in the original Oracle 9i, have made some necessary expansion according to its security needs. It is initiative and self-designed.

5 Safety Performance Analysis

With the application of security strengthening technology, the security of the database system is improved, but it also brings another defect, that is, to extend the database access to data and reduce the efficiency of SQL statements. In order to influence the database performance to the lesser extent, two methods are introduced: (1) optimize the security database settings, store the security database in a continuous storage space and design index to the database to speed up the query time; (2) when interface program is designed, optimize the accepted SQL statements to speed up access to the database and efficiency to carry out SQL.

Experimental results show that after using Oracle security strengthening device, only with less number of records then database access performance degrades as it still needs to go through the security authentication. Apart from that, the database data access performance is not significantly decreased, and security has greatly improved, especially nowadays when internet is so popular. It has broad prospects for development.

6 Conclusion

This paper analyzes the flaws and bugs that exist in user identification in Oracle 9i and designs the reinforcement security device. It also analyzes its interface design and database design, and designs a security command set to access the database, achieving the security reinforcement technology in terms of the user identity. Oracle 9i database management system with security reinforcement device can reach B1 security level.

This paper makes a study of the user ID in Oracle 9i. Aiming at the requirement of GA/T390-2002 in 6.3.3.1, it designs the authentication function of Oracle 9i database management system. The technology is innovative and practical to add the "shell" to Oracle 9i to reach B1 security level.

7 References

[1] Liu Qizhi etc. Web-based Oracle database security reinforced technology, Computer Science[J], 2003, 30(10): 379~382

[2] Bo Wenyang etc. Design and Implementation of Security Enhancements based on Oracle DBMS[J], Computer Science, 2001, 28(8): 327~330

[3] Xu Jiepan, The Investigation of data access security under Network Environment[J], Computer Science, 2002, 27(10): 267~271

[4] Huang Xiao etc. Authority Design and Implementation of Airport Oracle Database[J], Computer Engineering and Applications, 2004, 12: 184~187

[5] Wang Yongyi etc. Oracle Database Security Strategy[J], Anhui Vocational College of Electronics and Information, 2004, 3(5): 60~61

[6] Li Xiangjun etc, The Investigation of Oracle Database Application Security Problems[J], Nanchang University 2003, 25(4): 70~72

[7] Landwehr C E. Computer security[J]. International Journal of Information Security, 2001 8(1): 3~13

[8] National Computer Security Center. Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, Dec. 1985.

[9] The security classification standard of computer information system of the People’s Republic of China, China Standardization Press, 1999

10. Yang Anqi, Shan Zhenfang, Yang Jinlan. The Embedded SQL Program technology based on C/C + + and Oracle 9i[J], Control & Automation, 2005,2: 91~94

11. Binto George, Anna Valeva. A database security course on a shoestring[J]. ACM SIGCSE Bulletin, Proceedings of the 37th SIGCSE technical symposium on Computer science education SIGCSE '06, 2006.03,38(1): 7~11

A Circuit Simulation of Micro-Processor Model: using DSCH Software*

Minghuan Zhang [1]
[1]Department of Computer Science, Shanghai Sanda University, Shanghai, China P. R.

Abstract - *With popularizing computers and large-scale integrated circuits, designing of digital system has shifted from traditional manual model to circuit design using integrated circuit computer-aided design tools. That means hardware design is becoming software-oriented [1]. This paper designs a circuit of 4-bit digital micro-processor model by using logic simulation software DSCH, analyses functional mechanism of each module, and simulates its logic function.*

Keywords: microprocessor model, circuit simulation, DSCH

1 Introduction

A microprocessor incorporates most or all of the functions of a computer's central processing unit (CPU) on a single integrated circuit. Nowadays, microprocessors have been used everyplace, not only as the coral unit in microcomputers, but also as the key units in all kinds of digital smart devices.

Using DSCH, this paper is to design a 4-bit circuit of microprocessor model and to analyze functional mechanism of all modules. By simulating logic functions of all modules, the function of this 4-bit circuit of microprocessor could be achieved.

2 A 4-bit microprocessor model

2.1 The logic structure of a microprocessor

The logic structure of a 4-bit microprocessor is illustrated in figure 1, which includes such 8 functional modules as controller, program counter, program memory, accumulator, arithmetic unit and input/output register.

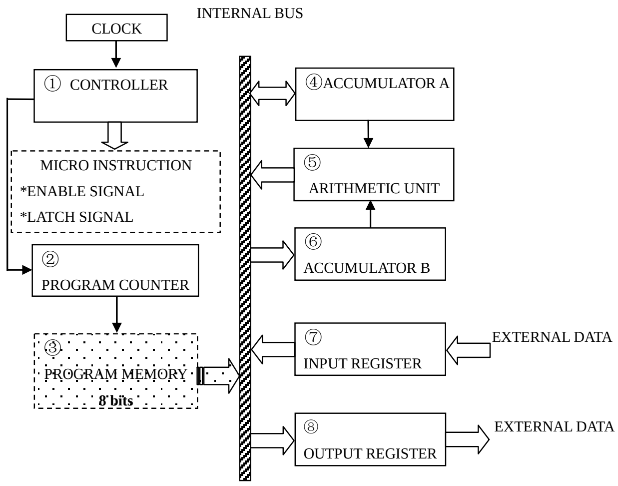


Figure 1 the function structure of a 4-bit microprocessor

2.2 Analyses of functional modules of the microprocessor

The eight functional modules of this microprocessor model communicate with each other via internal bus [3,4]. The results of analyses of eight modules are as followings:

① CONTROLLER is the headquarter coordinating the function of microprocessor. Based on the frequency of CLOCK, CONTROLLER sends all kinds of control signals directing the function of all modules.

② PROGRAM COUNTER is a 4-bit binary system counter, repeatedly counting from 0000 to 1111. After receiving an instruction from PROGRAM MEMRORY, PROGRAM COUNTER will automatically add 1 and get another instruction from PROGRAM MEMRORY.

③ PROGRAM MEMORY is a 8-bit RAM (capacity = 16 words ×8 bits), that is every word memories 8 bits

* The DSCH program is a logic editor and simulator designed by a research group under the leadership of E.Sicard, a professor of INSA of Toulouse [2]. DSCH is used to validate the architecture of the logic circuit before the microelectronics design is started. DSCH provides a user-friendly environment for hierarchical logic design, and fast simulation with delay analysis, which allows the design and validation of complex logic structures. The basic version is available online at http://www.microwind.net.

information. High 4 bits are instructions and low 4 bits are operands (e.g. augends or addend).

④ ACCUMULATOR A is a 4-bit register based on TRIGGER D. The register is used to memory temporarily variables of arithmetic operation (e.g. augends) or intermediate results. With the instruction of ENABLE A, ACCUMULATOR A sends data into internal bus.

⑤ ARITH METIC UNIT is to carry out arithmetic operation.

⑥ ACCUMULATOR A is a 4-bit register which memories secondary operands of arithmetic operation (e.g. addend).

⑦ INPUT/OUTPUT REGISTER is also a 4-bit register based on trigger D. INPUT REGISTER is to circulate external data to relevant functional modules via internal bus. The results are sent to external circuit by internal bus and OUTPUT REGISTER.

As showed in figure 1, microinstructions herein generated from CONTROLLER are simplified into two categories: ENABLE and LATCH. They sometimes generate impulse signals. For example, ENABLE A allows ACCUMULATOR A exchange information with internal bus; ENABLE Alu sends the operational results of ARITHMETIC UNIT into internal bus.

In the following part, the “button” is to be manually controlled to simulate the generation of microinstruction.

3 Simplification of microprocessor order sets

As aforementioned, within 8-bit PROGRAM MEMORY, high 4 bits are instruction codes, and low 4 bits are operands. Table 1 is used to illustrate how the instruction of addition “1+2” works.

Table 1 a simplified program (add operation)

Mnemonic codes	binary code	hexadecimal code
LDA 1 （1 Loaded into Memory A）	0101 0001	51H
ADD 2（Add operation A+2）	0001 0010	12H
OUT（results sent into external data）	0011 0000	30H
NOP（no operation, waiting）	0000 0000	00H

As showed in table 1, 0101 stands for the instruction LDA; 0001 for the instruction ADD, etc. Due to a large number of microprocessor instructions, mnemonic codes are often used in composing assembly language programs in order to facilitate memory.

Instructions are carried out on tap. When controller generates a signal, functional module will function on a particular timing sequence. Within microprocessors, controller in essence is a pulse-generated circuit concurring with the main clock, which generates trigger pulses on a particular timing sequence and directs the functions of relevant modules.

4 Logic simulation of microprocessor model

ACCUMULATOR A, PROGRAM COUNTER and ARITHMETIC UNIT are selected herein for analyses and simulation.

4.1 Analyses of the circuit of ACCUMULATOR A and its logic simulation

As demonstrated in figure 1, the logic functions of ACCUMULATOR A can be divided into two categories: sending operands to ARITHMETIC UNIT and exchanging data through internal bus with each register. The basic structure involves a 4-bit parallel INPUT/OUTPUT MEMORY based on TRIGGER D.

Figure 2 is a logic functional diagram using DSCH. Steps are as following:

① D-FF is high-level reset (clear 0). On beginning the simulation, the button CLEAR is set from 0 to 1 (after reverting phase, reset port RST of D-FF changes from 1 to 0), and then 4-bit output of register changes to Q=0；

② Outputting external data by hexadecimal keyboard (one hexadecimal number equals 4 binary numbers), then the data are loaded into port D via internal bus (0011 as showed in figure 2)

③ LATCH A controls inputs of synchronized clock. Setting LATCH A=1，Main CLK exports through NAND gate into common clock of register and then input status D of register is latched into output status Q;

④ 4-bit output Q of register is to conveyed into ARITHMETIC UNIT (module 5 in figure 1) marked as AluA3~AluA0 in figure 2;

⑤ Another port of register is under control of three-state gate. When ENABLE A=1 4 bits numbers latched in register nQ will export after reversal phase, marked as IB3~IB0;

Using demand "schema to new symbol" within DSCH, the diagram showed in figure 2 will change into an accumulator unit with 8 input ports and 8 output ports, among which the basic logic circuits are covered in a dark box.

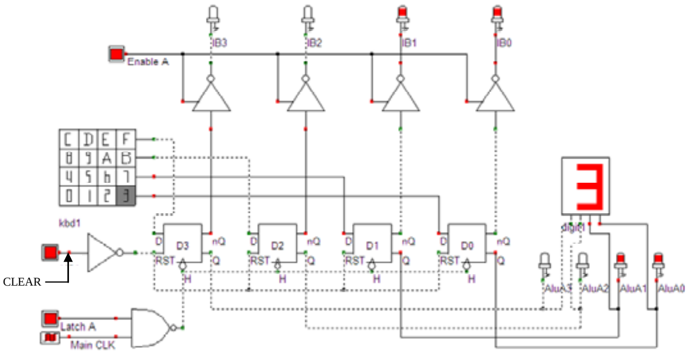


Figure 2 a logic simulation diagram of ACCUMULATOR A

4.2 Analyses of the circuit of PROGRAM COUNTER and its logic simulation

Presuming the microprocessor model used herein has 16 instructions, PROGRAM COUNTER is 4-bit binary code, 0000→1111 repeatedly counting by adding 1, used as the word address of 8-bit PROGRAM MEMORTY. Figure 3 is an asynchronous counter constructed with a 4-bit TRIGGER D. Each nQ port of D-FF connects D. Changes of output Q of low-level TRIGGER are used as inputs of high-level TRIGGERS. Increasing ENABLE COUNT and clearing o port of CLEAR. When ENABLE COUNT = 1, low-level of D-FF introduces clock pulse and begins counting.

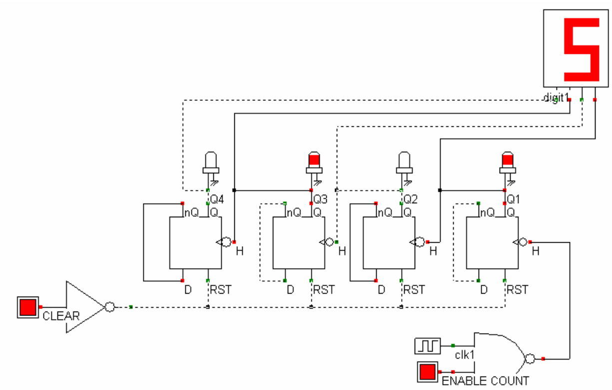


Figure 3 a logic simulation diagram of PROGRAM COUNTER

4.3 Analyses of the circuit of ARITHMETIC UNIT and its logic simulation

In this part, ADD and SUB operations will be carried out within a 4-bit ARITHMETIC UNIT and operands comes from ACCUMULATOR A and ACCUMULATOR B as showed in figure 4.

Analyses are concluded as followings:

① The main components of this circuit include a 4-bit full adder circuit, INPUT A (augends), INPUT B (addend compliments), INPUT C (carry bits from low-level), OUTPUT S（sum）and Carry (carry bits to high-level)

② Keyboard A simulates augends from ACCUMULATOR A; keyboard B simulates addends from ACCUMULATOR B;

③ The circuit calculating two's complement is constructed with 4 exclusive-OR gates and control port "addsub". When addsub=0, the circuit completed add operation; when addsub=1, the circuit completed subtract operation.

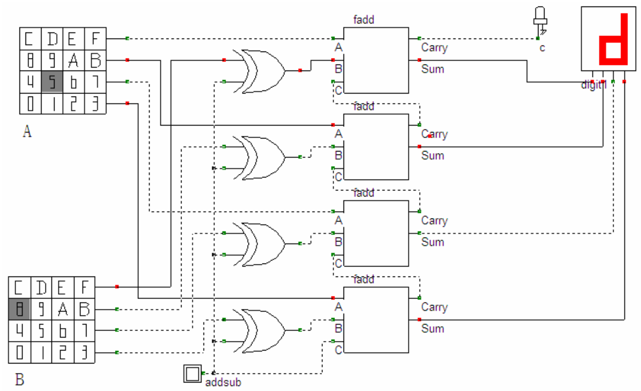


Figure 4 a logic simulation diagram of ARITHMETIC UNIT

For instance, keyboard A inputs 4 binary numbers, which are to be sent to full adder A as addends (noting: A is set as positive and its maximum is 0111). Setting AddSub=1, keyboard B exports 4 binary numbers sent to full adder B; when carry bit from the lowest level C = 1, input of addends in the form of complement is to achieve, that is (S)complements=A+(B)complements. The output of full adders should be the sum in the form of complement. sign bit S3=0 means the sum is positive, S true form =S complement ; sign bit S3=1 means the sum is negative, showing the sum in the form of complement

As showed in figure 4, if A=5（0101）,B=8（1000）, addsub=0, when add operation finishes, the result will be 1101 (D).

5 Conclusions

Thanks to the development of large-scale integrated circuits, the function of single circuits has increasingly complex and traditional manual design approach is to inevitably be substituted by automatic design. In this process, simulation software played a great role, bridging hardware designers and computer tools.

In teaching hardware courses such as digital circuit or principles of computer composition, use of simulation software could give students a direct and deep understanding, helping enhance teaching performance. The same is experience teaching. In addition to some experiences for hardware, use of simulation software is also helpful in intricate experience circuit.

6 References

[1] Donald A.Neamen. "Analyses and design of digital circuit" (third edition), (translated by Hongbao Wang). Tsinghua University Press, 2007.

[2] Jianwen Huang, Minghuan Zhang. "Basis of Modern Digital Circuit". China Machine Press, 2010.

[3] E.Sicard *et al*. "Advanced CMOS Cell Design". McGraw-Hill,2007.

[4] E.Sicard *et al*. "Basic of CMOS Cell Design". Tata McGraw-Hill, 2005.

7 Acknowledge

The author is indebted to Professor Jianwe Huang for the generous help and comments on an earlier version of this article.

Simulation Study of eLearning Classroom using iPads Based on Wireless LAN with IEEE 802.11b

Matthew Young Kim and Victor Clincy
Kennesaw State University
Kennesaw, GA, USA

Abstract - *This paper presents a simulation study of eLearning classroom using iPads as mobile devices based on the wireless LAN with IEEE 802.11b protocol. Moreover, this paper presents a simulation study whether the IEEE 802.11b can support up to 25 iPads of the eLearning classroom without fairness problems on a shared Wireless LAN. The simulation is performed by OPNET IT Guru Academic Edition 9.1.*

Keywords: Wireless Lan(WLAN), OPNET 9.1, IEEE 802.11b, iPads

1 Introduction

The iPad was selected by Time Magazine as one of the 50 Best Inventions of the Year 2010.[2] while Popular Science chose it as the top gadget.[3] behind the overall "Best of What's New 2010" winner Groasis Waterboxx.[4] The iPad has already several uses in the classroom, and has been praised as a valuable tool by many teachers. When the iPad first came out, Duke University created an application that provided students with a campus map, contact numbers, and other helpful information.

How can a professor utilize the iPad in their classroom? Well, a professor can keep their lecture notes on the iPad. They do not need to carry a ton of books and notes from class to class. The iPad can also be plugged into a projector. The professor, instead of having the students find something, can project the image onto a screen for the whole class to see. If a professor becomes sick, they can go back to past class's lectures and post them on the application, so the class do not fall behind schedule.

Wireless local area networks(WLANs), which are based on the IEEE 802.11 standard, are now very popular and one of the fast growing wireless access technologies in the world today. We can easily find many places such as cafes, bookstores, airports, and campuses.[5] WLANs provide an effective means of achieving data connectivity without constraints of physical wires in public places, offices, and campuses. The rest of the paper includes brief review to the background work, design and the results of the performance simulation environment, simulations evaluation experiments. Lastly, section 6 summarizes the paper and describes ongoing work.

2 Background Work

2.1 Web/Protocol

Three communication protocols, which are IP, TCP, and HTTP, are dominant to the today's web. The Internet Protocol (IP) is the principal communications protocol used for relaying datagrams (packets) across an internetwork using the Internet Protocol Suite. Responsible for routing packets across network boundaries, it is the primary protocol that establishes the Internet. IP is the primary protocol in the internet layer of the Internet Protocol Suite and has the task of delivering datagrams from the source host to the destination host solely based on their addresses.[9] For this purpose, IP defines addressing methods and structures for datagram encapsulation.

The Transmission Protocol(TCP) is complementing the IP, and therefore the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. TCP is the protocol that major internet applications rely on, applications such as the World Wide Web, e-mail, and file transfer. The FTP, HTTP, and Telnet are all examples of applications that require a reliable communication channel.[9] TCP is utilized extensively by many of the Internet's most popular applications, including the World Wide Web (WWW), E-mail, File Transfer Protocol, Secure Shell, peer-to-peer file sharing, and some streaming media applications. TCP is optimized for accurate delivery rather than timely delivery.

Lastly, the Hyper Text Transfer Protocol(HTTP) functions as a request-response protocol in the client-server computing model. In HTTP, a web browser, for example, acts as a client, while an application running on a computer hosting a web site functions as a server.[9] The HTTP protocol is designed to permit intermediate network elements to improve or enable communications between clients and servers. High-traffic websites often benefit from web cache servers that deliver content on behalf of the original, so-called origin server to improve response time. HTTP is an Application Layer protocol designed within the framework of the Internet Protocol Suite. The protocol definitions presume a reliable Transport Layer protocol for host-to-host data transfer.[2] The Transmission Control Protocol (TCP) is the dominant protocol in use for this purpose. In this simulation, HTTP is used to transfer web documents between eLearning server and iPads using HTTP/1.1.

2.2 IEEE 802.11b WLAN

IEEE 802.11b was proposed in 1999. It uses the 2.4GHz ISM band. There are several design objectives of IEEE 802.11b that make it more popular than IEEE 802.11a.[8] The main reason is that IEEE 802.11b is using the 2.4 GHz ISM band which is the same as that defined in IEEE 802.11. Together with the support of low data rate DSSS(1~2Mbps), backward compatibility is provided. This facilitates the interoperability of IEEE 802.11b and existing WLAN products on the market. [8] With relatively simpler modulation techniques, IEEE 802.11b devices are relatively cheaper. This technology provides low cost wireless Internet capability for end users, with up to 11 Mbps data transmission rate at the physical layer.

2.3 Wireless eLearning Performance

The wireless eLearning performance rely on the communication of the eLearning web clients(iPads) and eLearning web server. Sometimes, communication over the wireless network suffers from limited bandwidth, high error rates, and interference from another users in the shared wireless network channel.[10]

3 Simulation Environment

In this paper, I used OPNET IT Guru 9.1 Academic Edition for the simulations. OPNET IT Guru has certain advantages for the network design based on devices on the market, protocols, services and technology that is a trend in the telecommunications world.[1] OPNET IT Guru 9.1 assists with the testing and design of telecommunication protocols and networks by simulating network performance for wired or wireless network environments. The OPNET tool provides a hierarchical graphical user interface for the definition of network models as shown in Figure 1.a.[1] OPNET IT Guru comes with an extensive model library that includes the application traffic models(e.g HTTP, Email, Database, Video Conferencing) and protocol models(e.g IEEE 802.11b, TCP/IP, Ethernet).[1] Types of simulated services vary, be it the Internet(WEB), VoIP, file transfer, video conferencing, video streaming and others that can be set based on the needs of the user simulation. In general, OPNET IT Guru is sufficient as a reliable packet-based simulator.

The IEEE 802.11 WLAN architecture is built around a Basic Service Set(BSS). A BSS is all the devices associated with a local or enterprise IEEE 802. Wireless local area network(WLAN). With 802.11 it is possible to create an ad-hoc network of client devices without a controlling access point called an independent basic service set(IBSS).[6] In this mode, all mobile devices in the WLAN communicate with the server based WLAN.

4 Simulation Design

In this work, I used OPNET IT Guru Academic Edition 9.1 to simulate an eLearning classroom simulation scenario in Fig. 1.a. The network consisted of an iPad as a mobile client and eLearning Web Server as a WLAN server. The mobile client accessed web contents from the eLearning server using the IEEE 802.11b protocol(2.4GHz band). Node was represented an iPad. The WLAN connection operated at 1 Mbps, 2 Mbps, 5.5 Mbps, 11 Mbps automatically in the IEEE 802.11b, but I used only 11 Mbps in this simulation. The Wireless LAN parameters of the iPad were that the data rate(bps) was fixed with 11 Mbps. The Physical characteristic was the direct sequence, which represented each bit in the frame by multiple bits in the transmitted signal, instead of the frequency hopping and Infra Red.[7] It also had the channel settings with the 1,000 khz bandwidth and the 2.4Ghz minimum frequency. Lastly, the BSS identifier was fixed as 1.

The simulation factors had number of iPads(mobile clients) and web object size(light browsing, medium browsing, and heavy browsing) summarized in Table below.

Factors	Values
Number of iPads	1, 10, 25
HTTP web object size	16KB, 32KB, 64KB

5 Simulation Results

5.1 Experiment 1

The first simulation model was the single iPad scenario shown in Figure 1.(a). The simulation results were shown from the OPNET simulations of the network in Figure 1. Moreover, these simulations were performed using the HTTP/1.1.

5.2 Experiment 2

I performed the ten iPads scenario to see whether each iPad got fair connections between the WLAN workstations(mobile clients) on a shared Wireless LAN. I also tested different size of objects(16KB, 32KB, and 64KB) in the experiment. Figure 2 showed the simulation results from this experiment of the ten iPads scenario.

Figure 2.(a) showed the ten clients infrastructure WLAN. Figure 2.(d) showed the wireless delay for the iPad client which was simulated for 12 minutes with time average. For high load condition of the high browsing(64KB), the delay was reaching over 0.01 sec.

Basically, this experiment was performed under the HTTP/1.1 instead of the HTTP/1.0, so this got more advantages of the TCP acknowledgement packet. Figure 2.(e) showed the WLAN throughput(bits/sec) in the network level. Figure 2.(f) showed the average eLearning server response time at the node iPad1. With the 64KB load, the response time was almost 0.35 sec which was very close to the response time at the node iPad5(Fig 2.(g)) and iPad8(Fig 2.(h)). This indicated that there were no fairness problems between ten iPads on a shared Wireless LAN. That was the ten iPads shared the channel very fairly in the high load case(64KB).

5.3 Experiment 3

My primary question was whether the large numbers of iPads like twenty-five here can be supported in the 802.11b Wireless LAN. This was about the ability of a network to increase total throughput under an increased load when more iPads were added. Figure 3.(a) showed a large classroom simulation scenario with the same loads of the previous experiment 2. Figure 3.(e) showed the network level throughput results. Even with high load condition(64KB), the throughput was stable at about 210,000 bits/sec. Figure 3.(f), (g), (h), and (i) showed the HTTP response time at each node iPads, and they did not have fairness problems between twenty-five iPads on a shared WLAN. The network level throughput is higher than the application layer throughput because of protocol overhead such as TCP, IP, headers and retransmissions.[9]

6 Conclusions

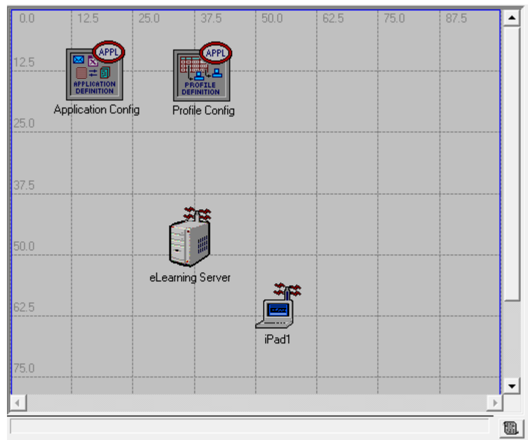
In this experiment, I simulated the performance of three different modeling scenarios with using 1, 10, and 25 iPads in the eLearning classroom. The simulation results showed that the IEEE 802.11b Wirelss LAN could successfully support up to twenty-five iPads no matter of conditions of light, medium and heavy web browsing without the fairness problem. For the next steps, I am going to simulate larger models of the eLearning classroom over 50. Furthermore, ongoing work is focused on the more realistic workloads such as video conferencing and video streaming.

7 References

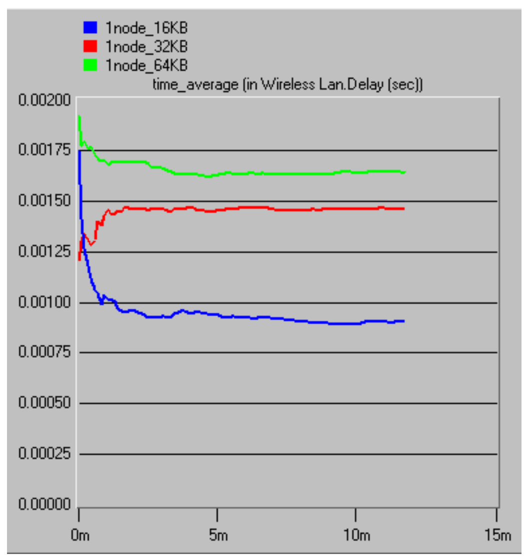
[1] OPNET Technologies, www.opnet.com
[2] Harry McCracken. "The 50 Best Inventions of 2010: iPad". Time Magazine, Nov 2010.
[3] Best of What's New 2010: Gadgets Popular Science. Dec 2010.
[4] Jannot, Mark. Best of What's New 2010: Our 100 Innovations of the Year Popular Science, 16 December 2010.
[5] T. Hansen, P. Yalamanchili and H-W. Braun, "Wireless Measurement and Analysis on HPWREN", Proceedings of Passive and Active Measurement Workshop, Fort Collins, Co, pp. 222-229, Mar 2002.
[6] 802.11 network, Wikipedia, http://en.wikipedia.org/wiki/
[7] Larry L. Peterson, Bruce S. Davie, "Computer networks: a systems approach 3th Ed.", Elsevier, pp. 132, May 2003.
[8] Yu-Kwong Ricky Kwok, Vincent K. N. Lau, "Wireless Internet and mobile computing: interoperability and performance", Wiley-IEEE, pp. 275, 2007.
[9] Books, LLC, "Internet Protocol", General Books LLC, pp. 43, 2010.
[10] D. Kotz and K. Essein, "Analysis of a Campus-Wide Wireless Network", Proceedings of ACM MOBICOM, Atlanta, GA, Sep 2002.

8 Appendix

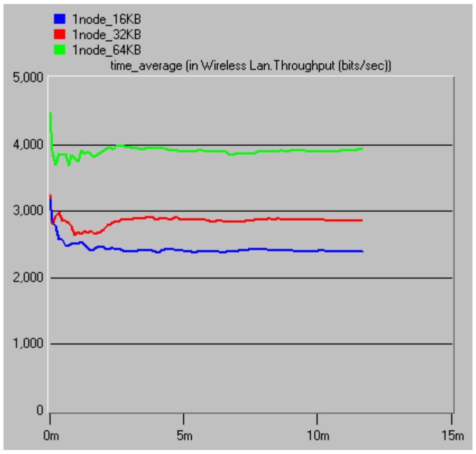
8.1 Experiment 1: 1 iPad client



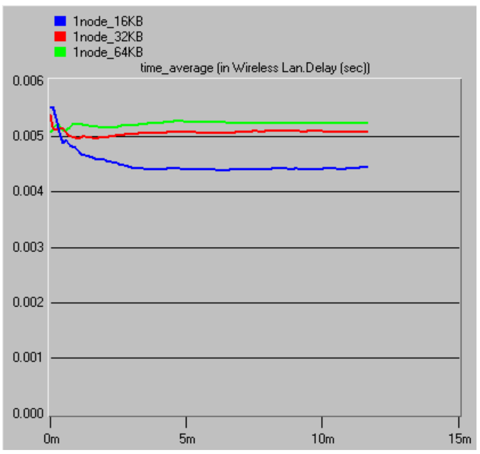
<**Fig 1.a** iPad classroom simulation scenario>



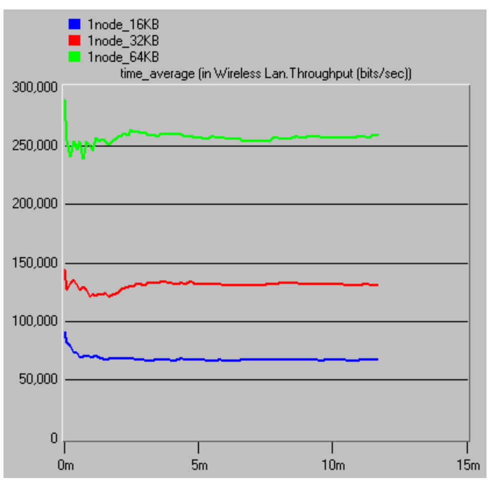
<**Fig 1.b** eLearning Server; WLAN delay(sec)>



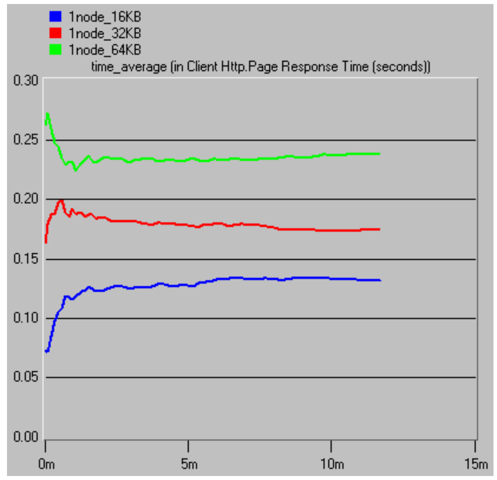
<**Fig 1.c** eLearning Server; WLAN throughput>



<**Fig 1.d** Average of iPad; WLAN delay(sec)>

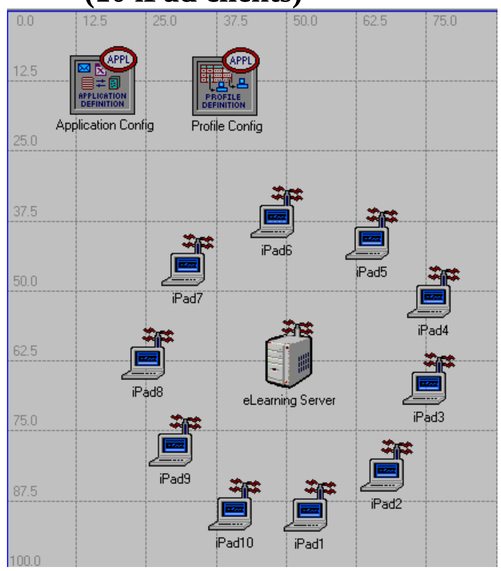


<**Fig 1.e** Average of iPad; WLAN throughput>

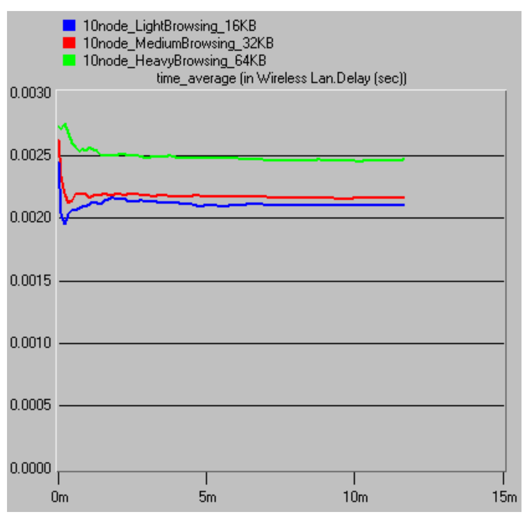


<**Fig 1.f** eLearning HTTP server response time>

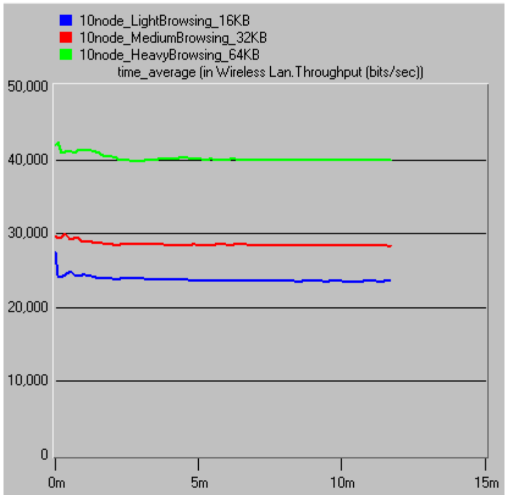
8.2 Experiment 2: Small Classroom (10 iPad clients)



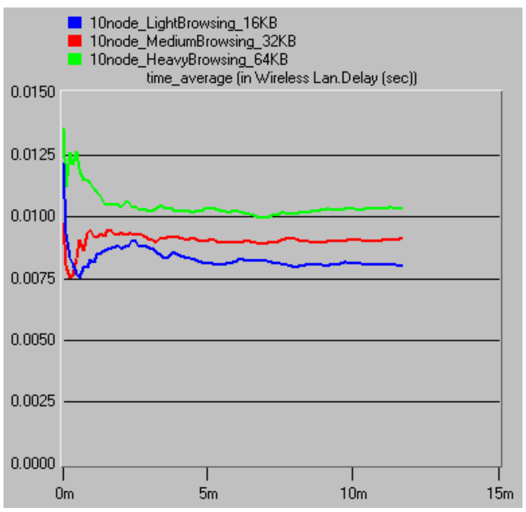
<**Fig 2.a** Small classroom simulation scenario>



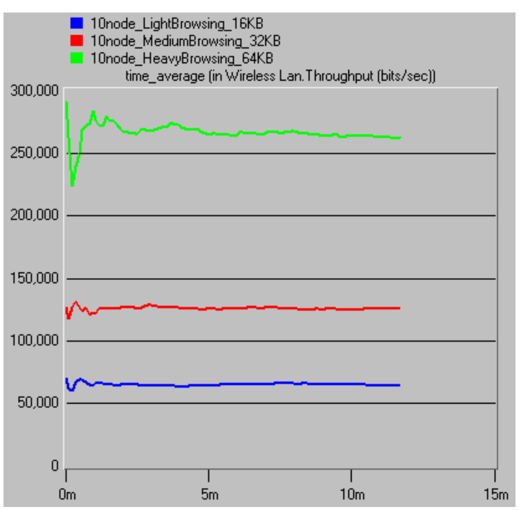
<**Fig 2.b** eLearning Server; WLAN delay(sec)>



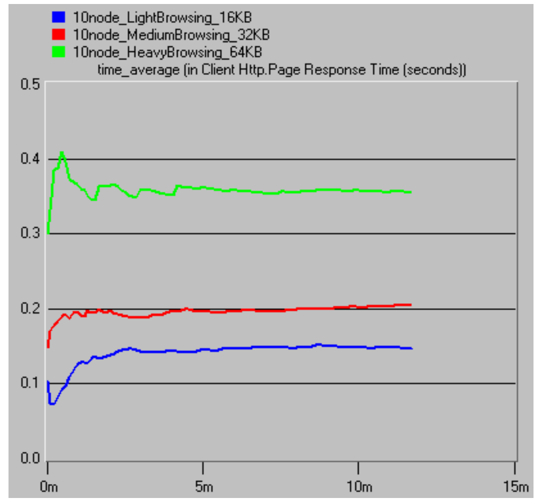
<Fig 2.c eLearning Server; WLAN throughput>



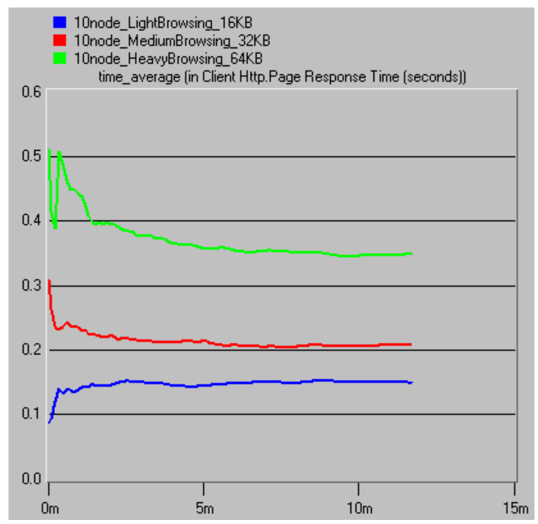
<Fig 2.d Average of iPad; WLAN delay(sec)>



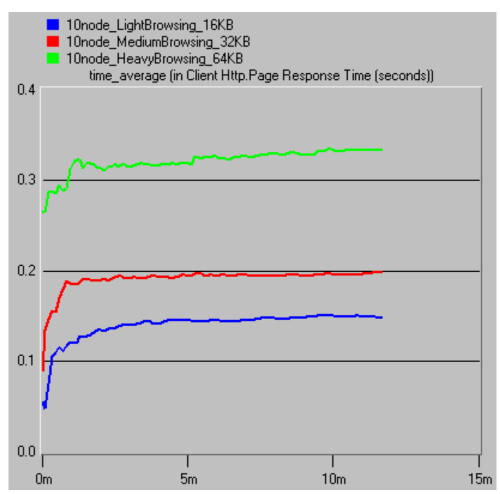
<Fig 2.e Average of iPad; WLAN throughput>



<Fig 2.f eLearning HTTP server response time(sec) at iPad1>

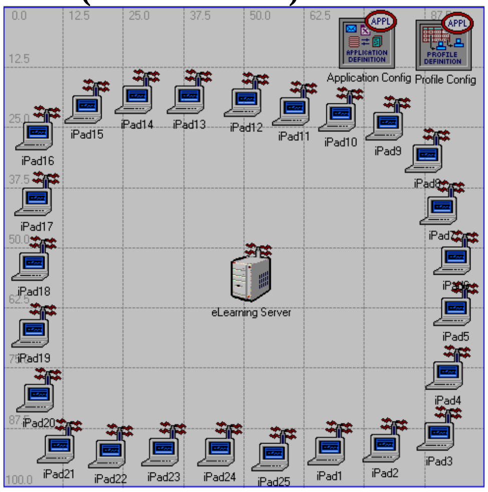


<Fig 2.g eLearning HTTP server response time(sec) at iPad5>

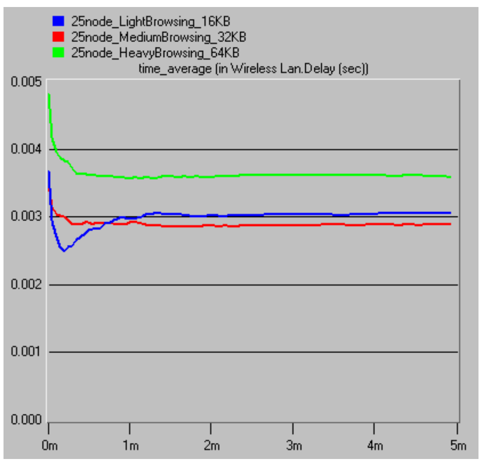


<Fig 2.h eLearning HTTP server response time(sec) at iPad8>

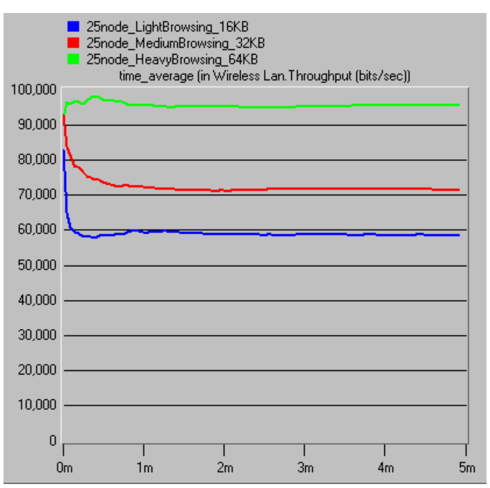
8.3 Experiment 3: Large Classroom (25 iPad clients)



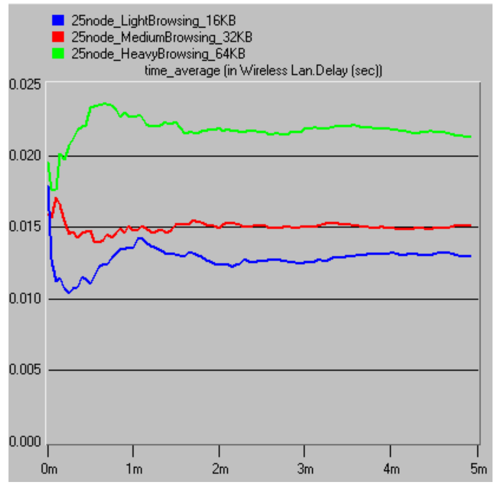
<**Fig 3.a** Large classroom simulation scenario>



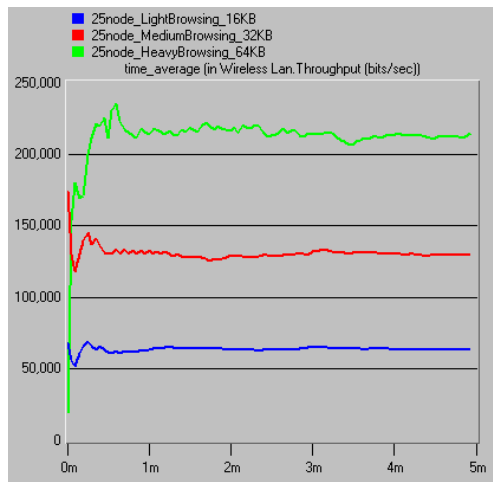
<**Fig 3.b** eLearning Server; WLAN delay(sec)>



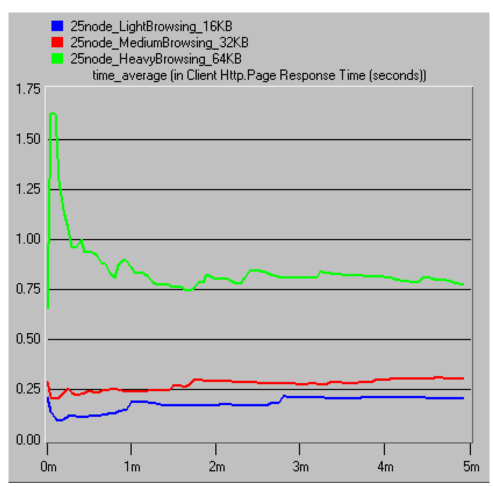
<**Fig 3.c** eLearning Server; WLAN throughput(bits/sec)>



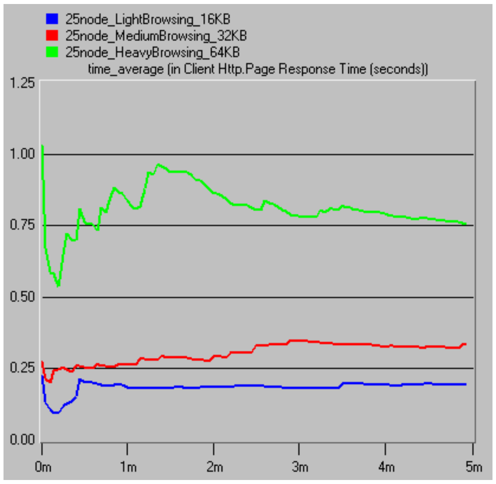
<**Fig 3.d** Average of iPad; WLAN delay(sec)>



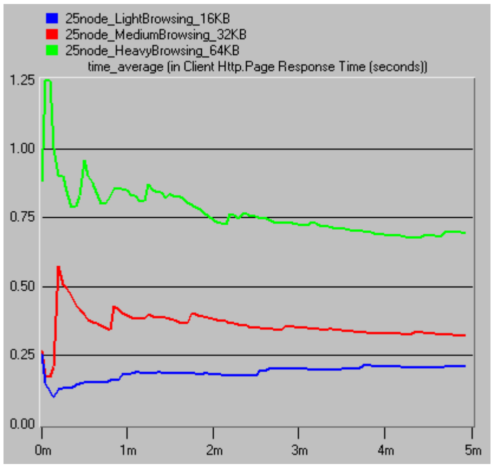
<**Fig 3.e** Average of iPad; WLAN throughput(bits/sec)>



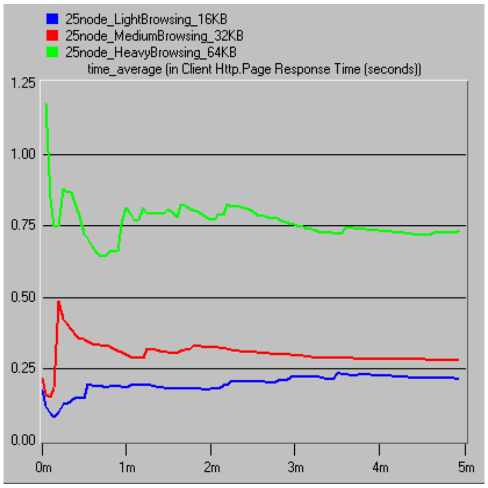
<**Fig 3.f** eLearning HTTP server response time(sec) at iPad5>



<**Fig 3.g** eLearning HTTP server response time(sec) at iPad12>



<**Fig 3.h** eLearning HTTP server response time(sec) at iPad18>



<**Fig 3.i** eLearning HTTP server response time(sec) at iPad24>

An approach for fast adder design

P.K. Lala
Department Electrical Engineering, Texas A&M University, Texarkana, Texas, USA

Abstract - *All digital signal processing systems are required to perform the basic arithmetic operation of addition with minimal delay. This paper presents a technique for fast addition operations that significantly reduces the propagation of carry signals, thus speeding up the addition process. In the proposed technique the carry value resulting from an addition is assigned a value before the addition, and then modified later if necessary*

Keywords: Carry propagation, 2's complement representation, Addition by column, Manchester adder, Carry skip addder

1 Introduction

An essential requirement of all digital signal processing systems is the ability to perform arithmetic operations with minimal delay. The speed of addition operation in such systems has a critical impact on its overall performance Over the years many techniques have been proposed for fast adder design [1]. The fundamental objective of all fast adders is to propagate the carry signal much faster than what it takes for the signal to ripple through in a ripple carry adder. An *n*-bit ripple carry adder is composed of *n* full adders with the carry-in of a full adder connected to the carry- out of its preceding adder.

The carry-select adder [2] design techniques uses two ripple carry adder segments , one of which has a 0 carry-in while the other has a 1 carry-in. Both segments operate in parallel. When the actual carry-in signal appears, a set of multiplexers is used to correctly select the sum of one of the parallel segments. The resulting carry-out signal is also correctly selected and propagated to the next carry-select adder segments. The carry-lookahead [3] or its variation Manchester adder [4], solves the limiting factor of ripple carry adders by calculating the carry signals from the input signals, before the sum bits are produced. The result is a reduced carry propagation time. A carry-skip adder [5] bypasses groups of consecutive adder stages if the carry propagation signals of all full adders in a block are at logic 1 i.e. it is known that carry will propagate through the block, thereby significantly reducing the carry propagate delay. The ripple carry adder has a delay of $O(n)$ for an *n*-bit adder. The carry look-ahead adder has $O(logn)$ delay, whereas both the carry-skip adder and the carry-select adder have $O(\sqrt{n})$ delay [6].

This paper presents an approach for fast addition of multi-digit numbers; the numbers can be all positives or a combination of positives and negatives. The numbers are assigned binary values in a way that allows estimation of the carry bits before the actual addition of the numbers. Moreover the traditional carry propagation process is drastically reduced..

2 A technique for addition

The proposed approach is different from the traditional way of sum generation; each number in a column of decimal digits is replaced first as indicated in Table I. Notice that the numbers 1 to 5 are represented by their binary values. The rest of the numbers i.e. 6,7, 8, 9, however first subtract 10 from itself; the resulting numbers -4, -3, -2 and -1 respectively are replaced by their 2's complement binary representation.

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	1100
7	1101
8	1110
9	1111

Table I Conversion of positive decimal numbers to binary

Before formally presenting the proposed summation algorithm let us illustrate the carry and the sum generation by performing the following addition:

$$5 + 3 + 4 + 8 + 6$$

The decimal values are first replaced by their binary equivalents using Table I.

To illustrate the proposed summation technique carry value generation and revision

5	0101
3	0011
4	0100
8	1110
6	1100

Two of the numbers 8 and 6 are converted to negative numbers as indicated by the 2's complement representation of the numbers. So the initial carry bits are assigned the binary equivalent of +2 i.e. 0010. The sum bits are 00110 i.e +6. In order. Thus the total sum of the column is 26.

Notice that 6,7,8, 9 are represented by two's complement of 4,3,2 and 1 respectively. Numbers 4, 3, 2 and 1 are obtained by subtracting 6,7,8,and 9 respectively from 10. Using the above concepts an algorithm for adding numbers that are all positive is given below:

<u>Algorithm I</u>:

i. Replace each number in a column by its binary equivalent using Table I

ii. Perform the addition of each column ; all the columns can be added simultaneously since the carry output from each column is considered only at the last step of the addition

iii. If the sum is a negative number then add 10 x k to the sum where k =1,2, 3…..depending upon whether the |sum| < 10, ≥ 10 but < 20, ≥ 20 but < 30 respectively, and subtract k from the carry. The carry and the sum bits together give the sum of the addends, else

iv. Add the carry bits of each column to the sum bits of its left column. This operation can be carried out simultaneously for all columns.

v. The resulting carry and sum bits are added as in step iv.

vi. Repeat step v till no carry bits are generated during a summation. The resulting sum is the final result of the addition

To illustrate Algorithm I let us perform the following addition

672
583
951
778
635

The replacement of the digits in each column as indicated in Table I results in the following:

6 1100	7 1101	2 0010
5 0101	8 1110	3 0011
9 1111	5 0101	1 0001
7 1101	7 1101	8 1110
6 1100	3 0011	5 0101
0100 1001	0011 0000	0001 01001

Since the result of the addition of the first column is 1001(-7), 01010(+10) is added to this, and 0001 is subtracted from the carry bits 0100 as below:

0100	1001
− 0001	+ 0 1010
0011	0011

The final result of the addition is derived as follows

First col.		Second col.		Third col
0011 0011		0011 0000		0001 01001
0011	0110		0001	01001

Replacing each 4-bit binary number by its equivalent decimal value results in sum of 3619 for the above addition.

The addition of all negative numbers can be carried out as discussed previously for positive numbers , the final result will obviously be negative. The addition of negative numbers can also be carried out by replacing each number with its binary equivalent as shown in Table 2. This table is obtained replacing -1 to -5 with their two's complement binary representation in the right column. The number -6 to -9 are first added to +10 and the resulting number are replaced with binary equivalent in the right column.

-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	0100
-7	0011

-8	0010
-9	0001

A combination of positive and negative numbers can be added as below.

Algorithm II :

i. Replace each number by its binary equivalent using Tables 1 and 2

ii. Derive the total positive numbers generated during the conversion process of the negative addends from decimal to binary using Table 2; let m be the number, and it is considered negative.

iii. Derive the total negative number generated during the conversion of the positive addends to binary using Table 1; let this number be n.

iv. Assign the sum (-m) + n to carry

v. Derive the sum of the addends

vi. If the sum is a negative number then add 10^k to the sum where k =1,2, 3…..depending upon whether the |sum| < 10, ≥ 10 but < 100, ≥ 100 but < 1000 respectively, and subtract k from the carry. The carry and the sum bits together give the sum of the addends, else

vii. Add the carry bits of each column to the sum bits of its left column. This operation can be carried out simultaneously for all columns.

viii. The resulting carry and sum bits are added as in step vii.

ix. Repeat this process till no carry bits are generated during a summation. The resulting sum is the final result of the addition

To illustrate let us perform the following addition.

[(-3) + 8 + 6 + 4 +5 + (-1)]

Using Tables 1 and 2 the decimals digits are converted to binary as shown below Since two positive digits, 8 and 6, are converted to negative, $n = 2$. Only one negative digit (- 1) is converted to positive binary number, so $m = -1$

	-3	1101
	8	1110
	6	1100
	4	0100
-	5	0101
	-1	01001

Hence the carry value = 2 + (-1) = 1, thus the

carry bits = 0001

The addition of the binary numbers resulted during the conversion process, results in

sum bits = 01001

Thus the actual sum of the of the addends is the concatenation of the sum and the carry bits

0001 01001

i.e. 19_{10}

As an another example let us perform the following addition of multidigit positive and negative numbers

575
- 328
101

We add the numbers in each column separately; note that the second digit in each column is considered a negative number. The numbers and their binary equivalents (derived using Tables 1 and 2) are shown below:

5	0101	7	1101	5	0101
-3	1101	-2	1110	-8	0010
1	0001	0	0000	1	0001

-

We would like to mention that all columns can be added simultaneously, since the derivation of the sum of any column does not depend on the carry out of the previous column. The sum and the carry bits for each column are shown below:

5	0101	7	1101	5	0101
-3	1101	-2	1110	-8	0010
1	0001	0	0000	1	0001

carry bits = 0000 carry bits = 0001 carry bits = 1111

sum bits = 0011 sum bits = 1011 carry bits = 01000

Next we use step vi of the addition procedure II. Since $k = 1$ we add 10_{10} (01010) to the sum bits of the second column,

and subtract 1_{10} (i.e. add 1111) to the carry bits. Thus the carry and the sum bits of the second column change to

carry	sum
0000	0101

Then the combined sum of the addition is :

First col.		Second col.		Third col.	
0000	0011	0000	0101	1111	01000

↘ ↙ ↘ ↙

0000 0011 0100 01000

Replacing each 4-bit binary number by its equivalent decimal value, the total sum is 348

3 Conclusions

This paper presents an approach for fast addition of fixed-point numbers. Two separate variations of the addition operation are proposed – one for positive numbers only, and the other one for mixed positive and negative numbers. These allow addition of multi-digit numbers, with anticipated carry value being assigned for each column of numbers before the actual summation operation .The assigned carry values are revised later, if necessary, based on the actual sum value obtained. The addition of all columns can be performed simultaneously , and the carry values are utilized only in the final stage of the addition. Thus the traditional carry propagation process is drastically reduced, thereby significantly improving the speed of adder operation

4 References

[1] I. Koren, *Computer Arithmetic Algorithms*, A. K. Peters, Natick, Mass, USA, 2nd edition, 2002

[2] O. J. Bedrij, "Carry-select adder", IRE Transactions on Electronic Computers, vol.EC-11, June 1962, pp. 340-346

[3] O.L. MacSorley, "High-speed arithmetic in binary computers." in IRE Proceedings. vol. 49, January 1961, pp. 67 – 91

[4] T. Kilburn, D. B. G. Edwards, D. Aspinall, "Parallel addition in digital computers: a new fast "carry" circuit", Proceedings of IEE, Vol. 106, pt. B, p. 464, Sept. 1959

[5] M. Lehman and N. Burla, " Skip techniques for high speed carry-propagation in binary arithmetic units", IRE Transactions on Electronic Computers, vol.EC-10, no.4 Dec. 1961, pp. 691-698

[6] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," IEEE Transactions on Circuits and Systems II, vol. 43, no. 10, 1996, pp. 689–702, 1996

Acknowledgement : This work was supported in part by the National Science Foundation, USA under Grant 0925080

SESSION

PROGRAMMING, MANAGEMENT, E-LEARNING, ETHICS, TOOLS, GUI, EVALUATION, CASE STUDIES, AND VISUALIZATION METHODS

Chair(s)

Prof. Hamid R. Arabnia

Contract Cheating
"The Hidden Trend in Computer Science Education"

Georgios Koumantaris
Computer Science Department, City University of New York - The Graduate Center, New York, NY, USA

Abstract *- Contract cheating is a new form of plagiarism that affects higher education everywhere in the world. A student can submit an original solution to an assignment by contracting an individual anywhere in the world to do the work and then present it as their own. There is a multitude of websites on the internet offering their services for a minimal fee. Computer science students hand in assignments they should of spent hours completing without learning any of the assignment objectives. This paper explains what contract cheating is and offers a plan to deter it in the classroom. Contents of this paper are the individual research of the author as a grader and lecturer for a period of thirteen semesters.*

Keywords: contract cheating, plagiarism, education, coding, auction sites.

1 Introduction

Plagiarism has evolved; students no longer copy each others' assignments. The difference between the traditional form of plagiarism and this updated version is that the work the plagiarizers are now submitting is unique – not a copy of someone's published work. Contract cheating falls within the category of plagiarism since students submit another person's work as their own.

Students now auction off their assignments on websites to various bidders. A student can post the assignment their professor has set onto an auction type website where various people bid on obtaining the contract to produce the assignment. The student then selects who they will enter into a contract with by reviewing reviews of the bidders' past work and by taking into consideration their asking price. Students aim at selecting the cheapest bidder with the highest reviews or experience. Most contracted freelancers are experienced computer programmers who sometimes work in groups. There are also ratings that help the student pick a superb and experienced coder, not just an average coder.

On various auction sites students merely pay the website and their contracted freelancer a small fee and they are awarded an A grade on 'their' assignment by their professor. These auction type websites act as the middle man holding the student's money until the chosen bidder has completed the assignment. Once the website has received the funds it announces this to the chosen bidder through their website. When the chosen bidder or contracted freelancer receives this information from the website they are assured they are going to be paid when the assignment is completed and hence they commence their work.

2 Example

Using a popular contract cheating site the phrase "school project" was searched and four results were given back. One of the search results was for a spell checker user interface:

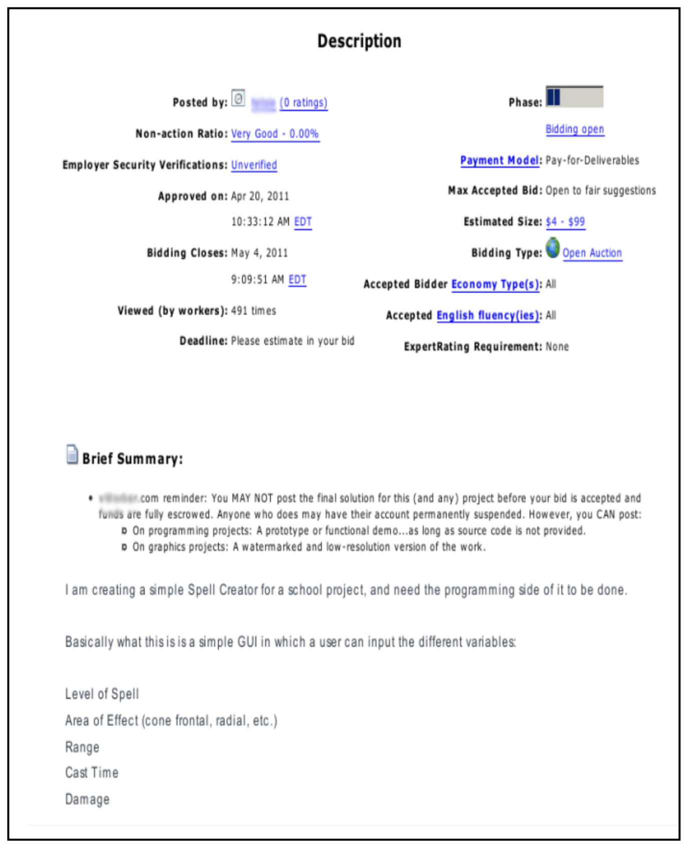


Figure 1 – Contract Cheating example

The above project (Figure 1) was viewed 491 times in a time period of 6 days. The student accepts bids more than $4 and less than $99, and wants to pay one fee for the final deliverable. The student's other option is to pay per hour. Another option is the type of economy of the coder's origin. The student can choose a coder from an emerging economy (China, Romania) or a mature economy (UK,USA). A coder from a mature economy will be 200-300% more expensive than a coder from an emerging economy according to the contract cheating website. This particular student has no preference on the type of the economy as long as the coder is within the asking price range. Below the description of the project there is a brief summary of the project. Furthermore, there is additional material attached on the project for the freelancer coder to examine.

3 Backgound

Auction websites are legitimate organizations that help many individual programmers and companies sell their services. This paper does not condemn the existence of these websites. Since they are profit driven, they just cannot ignore a big part of the software services market which is university students. Thomas Clarke and Robert Lancaster (2006) researched rentacoder.com (now renamed vworker.com) and concluded that one out of eight projects completed via the website, were university assignments [1]. After studying a two year period from 2004 to 2006 and 912 cases, Clark and Lancaster allocated half of these cases to the United States[2].

It does not cost any money to create an account on these software services websites, it is only a matter of having the time to do it. As an example of how inexpensive these services are we offer the example of www.crunchbase.com. This website is “a free database of technology companies, people, and investors that anyone can edit,” similar to a social networking website. It is a very valuable communication tool in the technology industry. The website was coded by a Bosnian/Yugoslavian Eastern European firm which was contracted through a software services website (rentacoder.com) for as little as $500.00 [5]. A student can contract a project for as little as $20.00, no matter how difficult or time consuming the project is.

Contractors are always from countries were the economic situation does not permit them to profit the same monetary amount within their own country. It is a monetarily profitable arrangement for both parties. It is rare to find a student from a mature economy like the United States contracting a coder also from the United States. For a student in the United States, spending sixteen US dollars on contracting a freelancer is extremely affordable. For a freelancer in an emerging economy like Romania for example, the sixteen US dollars is a significant amount of money.

3.1 Solution

Lancaster & Clarke established multiple solutions to remedy the situation [1] [2] [7]. One of the solutions proposed in one of their papers is to monitor auction sites. Lecturers can keep a list of websites and have their tutors monitor these sites. This sounds easy to do but multiple new websites appear every month therefore monitoring all these sites is not achievable. The Systematic Six-Stage Process for Detecting Contract Cheating also from Lancaster & Clarke is not feasible within a semester’s time, although it is a good countermeasure [7].

Students tend to hide their real information to avoid exposing their identity and websites never release personal information about their users. Rentacoder.com's owner, Ian Ippolito, said, "We don't have the manpower to manually review 100% of the postings, I actually spend considerable employee resources going through 50% of the postings. When we see a project that involves taking a test for another person we remove it and inform the person why." According to the United States' privacy laws, under the Federal Communications Commission, auction sites are forbidden from releasing students' details to universities [4].

The only way personal information in reference to the users can be released is to send a court order to the auction site demanding the release of the information [4]. This process would take months of work and preparation. This paper is proposing that the remedy to the problem of contract cheating must be fast and accurate. A professor must deal with the problem of contract cheating within the time frame the assignment is corrected and given back to the students. Unfortunately professors to not have a limitless amount of time at their disposal to verify a student handed in work which they contracted to another party.

Professors, who tested their students frequently, would know each student's programming capabilities therefore they would not have a problem identifying whether they programmed their assignments or contracted them out to a site. But generally this is not the case. Professors usually do not test their students as frequently as they would like to or even if they do, they are not the ones correcting the programming assignments. The remedy to this problem is an AI program to obtain a signature of all the old student's assignments and compare it to the signatures of a new assignment.

The solution can be implemented with careful resource allocation and pattern recognition of student’s assignments. Computer Science departments can gather and store students' work from all of their programming related classes. The department can create a programming project portfolio for each student. By studying the work found in a student's portfolio, the student's programming signatures can be obtained and documented for a professor to use. When a signature is documented, it can be cross-referenced by their professors with the student's new assignment's signature.

Programming signatures come in many different shapes and forms. Some students may have a signature within their comments; some may have a signature within their variables.

There are many different ways to identify signatures in programs. Object oriented languages like C/C++ are examined for code structure for example (function structure, variable definition/declarations). A student might go from pure C to only C++ coding standards from one assignment to the other. Student programmers in general differ from one another in terms of their style.

For example, the program will identify Hungarian notation from CamelCase Notation. In Hungarian Notation variable names are divided into two parts: the lowercase prefix, and the qualifier. The lower case prefix has the information about the variable type and the qualifier tells you what the variable holds. The variable pstrError for example is a pointer to a string that holds an error message. In CamelCase notation a variable is a compounding of phrases with no spaces, just separated by their starting capital letter. For example PoiStrErr in CamelCase would have the same meaning as pstrError in Hungarian Notation.

This algorithm goes through all of a student's previous work gathered, and identifies the similarities or signatures. Any observable patterns identified within students' programming project can also be added for future reference. There is no yes or no answer in regards to the question “whether or not the student created the assignment in question”. The program will give a list of patterns recognized within the suspicious assignment in question and the percentage of the similarity between the signature of the assignment in question and signature of all the past assignments. By comparing the percentage of similarity of programming signatures a professor can get a better idea of whether or not the assignment was contracted or not.

If a student's assignment is exceeding a certain threshold of similarity (controlled the professor) against the up-to-date signature, a second method of contract cheating discovery is implemented. A list of contract cheating websites can be crawled to identify a number of things. The student's user name, the assignment handout patterns, due dates of the project and posting dates. Building a crawler is a straight forward process that would not require each lecturer/grader to play detective every time there is suspicion of contract cheating. This software forensic tool (pattern recognition and crawler) will automate the finding process and present the results back to the instructor. After the results are given to the instructor it is up to them to pursue the student's conviction any further.

The software forensic tool can be very effective since students do not spend time to conceal evidence of contract cheating. Usually students submit projects the same way they receive them from their winning bidder. The reason they contract the work is that they have no idea how to do the work, therefore they are afraid to make any changes. Also another reason is that they have no time to work on the project, therefore they have no time to change the code.

4 Future Work

The software tool itself is not sufficient. In order for the successful implementation of the above work, Computer Science departments need to gather all of their students' code logs. A good future project is to create a web interface, where each university will create an account and the professors would submit their student’s work and create signatures to compare with new assignments. A second paper is on the way with type of signature and code examples. Also statistical research needs to be put together and summarize the accuracy of the method.

This long term research can be extremely useful to Computer Science Education all over the world. Benefits will spread across Computer Science Departments and Universities will produce healthy programmers for the industry.
Length

5 References

Robert Clarke and Thomas Lancaster (2006). “Eliminating the successor to plagiarism? Identifying the usage of contract cheating sites” in *Proceedings of the 2nd International Plagiarism Conference*, Gateshead, UK.

Lancaster & Clarke (2007-08-30). "Assessing Contract Cheating Through Auction Sites – A Computing Perspective" (PDF). http://www.ics.heacademy.ac.uk/events/8th-annual-conf/Papers/Thomas%20Lancaster%20final.pdf

Cheating students put homework to tender on Internet". Daily Mail/dailymail.co.uk. 2006-06-13.http://www.dailymail.co.uk/pages/live/articles/news/news.html?in_article_id=390317&in_page_id=1770. Retrieved 2006-06-14.

Jessica Shepherd. May 6, 2008 The Guardian, http://www.guardian.co.uk/education/2008/may/06/highereducation.students

http://www.startupgazette.com/2011/03/crunchbase-vs-gazettebase-crunchbase-was-a-rent-a-coder-project/

http://www.tenouk.com/cnotation.html

Thomas Lancaster and Robert Clarke (2007). “Establishing a Systematic Six-Stage Process for Detecting Contract Cheating”, in *Proceedings of ICPCA07* pp.342-347.

Improving Exception Messages with ExceptionDoctor

Michael Woods[1] and Stephen H. Edwards[1]
[1]Department of Computer Science, Virginia Tech, Blacksburg VA US

Abstract - *Beginning programmers often have difficulty interpreting exceptions and using the associated messages to pinpoint the cause of incorrect program behavior. When an interactive development environment (IDE) presents a novice developer with a runtime time exception, it generally provides with a stack trace and a limited, cryptic exception message that is hard for a beginner to interpret. This paper describes ExceptionDoctor, a Java utility that solves this problem. ExceptionDoctor intercepts exceptions thrown by student code and improves the embedded exception messages to provide level-appropriate descriptions. ExceptionDoctor also examines the source code that produced the exception (if available) in order to describe the immediate cause of the exception in student-level terms.*

Keywords: Java, run-time error, exception handler, explanation, debugging.

1 Introduction

As entry-level programmers begin learning Java, they are quickly and abruptly introduced to the runtime exception mechanism inherent to the JVM, or Java Virtual Machine. Without proper instruction, these novice developers are forced to fight their way through understanding these runtime exceptions without the understanding of stack traces and exception messages. To understand the messages that the JVM provides the user, the student must understand what causes that exception in the first place.

ExceptionDoctor rewrites Java runtime exception messages to be user friendly and more context sensitive.

ExceptionDoctor's improved exception messages include three pieces of information: the offending line of code (if available), an improved exception message that both explains the meaning of the exception and the most likely cause(s), and the original exception's stack trace. The improved exception message is derived from the original exception and contains information derived from the offending line of code such as how certain variables may have contributed to the exception's cause. The improved message is presented in paragraph form with ExceptionDoctor's best guess on which portions of the source code caused the problem, how the issue can be avoided, and the name of the exception printed at the bottom.

Another unique aspect of ExceptionDoctor is the redesigned architecture that allows for easy integration with existing IDEs and runtime exception handling programs. ExceptionDoctor can be added to handle any uncaught exceptions in a Java program by simply adding it to the classpath and using an additional command line argument when running the target program. The distributable library for ExceptionDoctor automatically inserts itself into JUnit tests [1], so that uncaught exceptions during testing receive benefits. If desired, ExceptionDoctor can be integrated into internal exception handlers (e.g., for "caught" exceptions) using a single line of code added to the targeted catch blocks. Finally, ExceptionDoctor can be added to projects easily in most interactive development environments (IDEs), including BlueJ [3] and Eclipse [2], so that students can see the new exception messages directly in the IDE's output. It can even be added onto automated grading tools.

ExceptionDoctor's design uses a data-driven dispatch structure to improve efficiency and increase flexibility. As a result, once can replace specific messages and extend the system with new messages for new types of exceptions, all without modifying or even recompiling the ExceptionDoctor code.

2 Relationship to Previous Work

While not many projects attempt to improve messages returned by Java exceptions, presenting the exception messages in a more readable format has been implemented by many IDEs.

BlueJ [3], like many other IDEs relies on the built-in messages generated by the JVM for runtime exceptions that occur in client code. These are presented in the terminal window in the same way they would on the console if the program were run stand-alone. For exceptions that arise inside JUnit tests, BlueJ attempts hides some of the more distracting exception information from beginning users, displaying the most important information when specifically requested. When an exception occurs during a JUnit test run in BlueJ, the test results window includes a box that displays the exception's message and a shortened stack trace. BlueJ's shortened stack trace only shows the lines of code written by the student that caused the exception. This allows a student to focus on the lines of code that threw an exception; however, it does not help the student learn what they did wrong.

ExceptionDoctor takes a different approach to displaying exceptions. While ExceptionDoctor helps students focus on the line where the exception occurs, it also attempts to teach them what part of the line caused the exception and how it might be avoided in the future.

ExceptionDoctor was inspired by and is based on the exception explanation features implemented in the Backstop project [4] created at Columbia University. Backstop provides Java execution tracing support together with a runtime exception handler to rewrite exception messages. While ExceptionDoctor has reused many of the error messages displayed by Backstop, the entire dispatch structure has been redesigned, and then packaged in a radically different deployment mechanism. While Backstop was only able to reinterpret exceptions thrown by programs passed directly to it, ExceptionDoctor is capable of integration with existing projects. This feature is an important difference between the two because it allows ExceptionDoctor to seamlessly integrate with existing professional tools with little effort (i.e., JUnit and Eclipse). By integrating ExceptionDoctor with existing tools, students are able to work in a professional environment with added training wheels. As a student progresses, they can easily remove the jar from the project and continue using the same tools.

3 ExceptionDoctor's Design

ExceptionDoctor is composed of two different structures. The first structure is the handler class. Handler classes allow ExceptionDoctor to wrap exceptions with an improved exception message. The second structure is the exception map. The exception map contains all of the mappings from exception classes to their corresponding handlers. The exception map is a singleton that is empty when initialized and filled dynamically as exceptions are wrapped and new mappings are found.

When ExceptionDoctor is passed an exception, it initially looks for an existing relationship in the exception map between the exception and a corresponding handler class. If no relationship exists, ExceptionDoctor looks through the class path for classes that could wrap the exception. If no classes exist, ExceptionDoctor then looks at the exception's superclass and attempts to wrap it. Because all of Java's exceptions descend from the Throwable class, which has its own default handler, there is a handler for every Java exception passed to ExceptionDoctor. Wrapped or nested Java exceptions are also handled by unwrapping first.

After a successful lookup has been performed, ExceptionDoctor will then cache the exception to handler relationship in the exception map and wrap the exception with the handler that was found during the previously mentioned search. Handler classes are only able to change the message that will be displayed to the user, ExceptionDoctor enforces that a wrapped exception still has the original stack trace and has the original exception as the cause.

3.1 Dispatch Structure

ExceptionDoctor includes a unique dispatch structure that allows for dynamic lookups of possible exception handler classes. When an exception is passed to ExceptionDoctor, it uses class names to determine the exception-to-handler relationship. All handlers must follow the naming convention ExceptionNameHandler, for example: IOExceptionHandler.

Using this naming convention, new exception handlers can be created in the project's workspace. When ExceptionDoctor is called, the new handler is dynamically found through a search of the classpath. Assuming this class implements the ExceptionHandlerInterface, the discovered class is mapped to the exception in the exception map and used to wrap the exception in all future references.

New exception handlers only need two things to be integrated with ExceptionDoctor, they must follow the previously mentioned naming convention and they must implement the ExceptionHandlerInterface. The interface ensures that the class has a wrapException(Throwable exception) method that returns a wrapped exception. The ExceptionDoctor API also includes an Abstract-ExceptionHandler that provides a set of utility functions for handler classes. These utilities help standardize the messages and exceptions that ExceptionDoctor produces.

3.2 Handlers

Every handler must implement the Exception-HandlerInterface and provide a wrapException() method that returns a newly wrapped exception. Each of the handlers can also extend the AbstractExceptionHandler, which provides methods for accessing the offending line of code.

As an example, the code in Figure 1 shows a code sequence that throws an ArrayIndexOutOfBounds-Exception. When this code is run, the appropriate Array-IndexOutOfBoundsHandler will catch the exception and rework the message from '-2' to the message that is displayed in Figure 2.

Figure 2 was captured from eclipse's console after the uncaught exception terminated the program. The wrapped exception retains all information from the original exception. This allows the student to see what a normal exception stack trace looks like.

3.3 Dynamic Caching

ExceptionDoctor implements a dynamic caching system in the map from exception classes to their handlers. Each time an exception is passed to ExceptionDoctor for translation, the exception map attempts to find a matching handler. If a handler is found for the passed exception, this results in a direct link in the exception map. This allows ExceptionDoctor to dynamically populate its mappings from exceptions to handlers as needed.

Dynamic caching allows for quick access to these mappings when batch processing or when integrating ExceptionDoctor into grading servers.

3.4 Running With ExceptionDoctor

While ExceptionDoctor integrates itself into JUnit's test execution infrastructure, one must explicitly ask for its services to receive improved exception explanations in a plain Java program.

Normally, a Java program can be run using a command like this:

```
java <MainClassName> <args> …
```

To include ExceptionDoctor, the command should be rewritten as:

```
java ExceptionDoctor <MainClassName> <args> …
```

This addition ensures that ExceptionDoctor will be able to access all uncaught exceptions. If ExceptionDoctor is unable to rewrite the exception message for any given exception, the original exception is passed through without change. Section 5 describes automatic configuration with JUnit tests.

4 Impact on Student Development

In recent research into the most common mistakes made by novice programmers, it was found that file I/O, null variable, and type errors were the most common exceptions to arise [5]. Each of these error types corresponds to a runtime exception that ExceptionDoctor handles.

```
int z = 3;
int y = 5;
Object x[] = new String[5];
Object k = x[z-y];
```

Figure 1: This code segment generates an exception.

```
java.lang.ArrayIndexOutOfBoundsException:
    In file DemoProgram.java on line 52, which reads:

    Object k = x[z - y];

    It seems that the code tried to use an illegal
    value as an index to an array.  The code was
    trying to access an element at index -2 of the
    array called "x".  Remember, you cannot have a
    negative index. Be sure that the index is always
    positive.  The variable "z - y" had the value
    -2 when the error occured.

    This error is called an
    ArrayIndexOutOfBoundsException.

    at Main.weeklyAverages(DemoProgram.java:52)
    at Main.computeWeatherStats(DemoProgram.java:45)
    at Main.main(DemoProgram.java:18)
```

Figure 2: Example exception message rewritten in Eclipse.

```
String fileName = "/home/mike/notreal/foo.bar";
File f = new File(fileName);
Scanner s = new Scanner(f);

...
java.io.FileNotFoundException:
    In file Junit3Test.java on line 15, which reads:

    Scanner s = new Scanner(f);

    It appears that the code was trying to operate on
    a file called /home/mike/notreal/foo.bar.
    However, it seems that this file may not exist.
    Check that the filename is spelled correctly.
    Analysis shows that /home/mike/ is a valid path.
    The remainder of the file is invalid.

    This error is called an FileNotFoundException.

        at Junit3Test.testNoFile(Junit3Test.java:15)
```

Figure 3: An example of a FileNotFoundException.

According to Jadud [5], the most common error type is file I/O. File I/O exceptions can be caused by novices attempting to open a file or write to a file. If a student is attempting to open a file, there are generally two states that the file can be in that will result in an exception. The first state is a nonexistent file. When the file does not exist, ExceptionDoctor checks the given path for errors in the directory structure. If the directory path is correct, then the user is simply told that the file does not exist. However, if the path is incorrect, it is more likely that there is a spelling mistake in the path. Figure 3 gives an example of output in this situation. File exceptions can also occur because of bad file permissions.

I/O errors are also caused by errors in writing and reading strings from input and output streams. However, IOExceptions can be complex exceptions. Because IOExceptions are so complex, ExceptionDoctor does include message support for every situation that could cause an IOException. However, through diagnostic tools such as ClockIt [6], an instructor can gather information about the types of exceptions that students are encountering and create a custom IO handler class to remind students of lessons they learned in class.

The next most common kind of exception is due to an uninitialized variable. An uninitialized variable most commonly manifests itself as a NullPointerException or various data structure exceptions. When NullPointer-Exceptions are thrown, the novice has created a variable that does not refer to any object. When ExceptionDoctor encounters this, it attempts to locate the null variable in the source code. If there is only one potentially null variable in the line, the explanation simply uses that variable name in describing the cause. If there are multiple nullable variables, a list of possible null variables is included in the message instead. Figure 4 shows output for the single-variable situation.

```
String nullString = null;
Integer i = 1;
Integer j = 2;
nullString.substring(i, j);

...

java.lang.NullPointerException:
    In file Junit3Test.java on line 22, which reads:

    nullString.substring(i, j);

    It appears that the code was trying to call a
    method or refer to a member variable on an object
    called "nullString", which is null.  Make sure
    the variable has been initialized in your code.
    Remember, declaring the variable isn't the same
    as initializing it.  You may need to initialize
    the object using the keyword "new".

    This error is called an NullPointerException.

        at Junit3Test.testNull(Junit3Test.java:22)
```

Figure 4: An example of a NullPointerException.

Exceptions in initialization can be caused by data structure variables outside of the declared bounds. The simplest data structure that suffers from this error is the array. When ExceptionDoctor encounters an ArrayOutOfBoundException, it attempts to narrow the exception cause down to three common mistakes. ExceptionDoctor can identify errors caused by negative array indexes, accessing an element in an unpopulated array, and using an array index that is greater than the array size.

```
Object[] tenElementArray = new Object[10];
Object foo = tenElementArray[15];

...

java.lang.ArrayIndexOutOfBoundsException:
    In file Junit3Test.java on line 27, which reads:

    Object foo = tenElementArray[15];

    It seems that the code tried to use an illegal
    value as an index to an array.  The code was
    trying to access an element at index 15 of the
    array called "tenElementArray".  The size of the
    array may be less than 15. Keep in mind that if
    the array size is N, the biggest index you can
    access is N-1.

    This error is called an
    ArrayIndexOutOfBoundsException.

        at Junit3Test.testBadIdx(Junit3Test.java:27)
```

Figure 5: An example of an ArrayIndexOutOfBoundException

ExceptionDoctor handles an ArrayIndexOutOfBoundsException with only the source code line that caused the exception and the value of the index. Using only these two pieces of information, ExceptionDoctor only gives the student hints on the exception's cause based on its best guesses about the array. Figure 5 shows an array with ten elements contained and an attempt to access element fifteen, a non-existent element.

Novice programmers also have issues with variable types. Variable type confusion would most likely surface in the form of a NumberFormatException. NumberFormatExceptions occur because of the different precision number types available in the Java syntax. Students might find difficultly when understanding the difference between an integer and a float. When ExceptionDoctor encounters a NumberFormatException, it attempts to parse the string multiple times using different number parsers to determine the actual format of the string. For example, Figure 6 shows the output from an attempt to perform an improper parse of a float as an integer. ExceptionDoctor attempts to parse the string as a float and reports the correct number format of the string as a float.

```
Integer.parseInt("3.59");

...

java.lang.NumberFormatException:
    In file Junit3Test.java on line 31, which reads:

    Integer.parseInt("3.59");

    It seems that the code wants to convert a String
    to an integer.  However, the String "3.59"
    appears to be a floating point value, not an
    integer.  You may want to use a different
    datatype to store the value, like float.

    This error is called an NumberFormatException.

        at Junit3Test.testParse(Junit3Test.java:31)
```

Figure 6: An example of a NumberFormatException.

5 Availability

One of the key features of ExceptionDoctor is its ease of integration into a variety of different tool chains using JUnit. ExceptionDoctor is distributed as a single JAR file. Once added into a project's classpath (ahead of JUnit, if JUnit is also being used), ExceptionDoctor will automatically configure itself. Uncaught exceptions in JUnit tests (3.x or 4.x) are enhanced without any additional effort required of the student. Enhanced exception messages appear in the IDE's normal output mechanisms, whether that is a simple console window or a fancier GUI results pane (ex. Eclipse), since ExceptionDoctor does not produce any output of its own and simply redecorates the message contained within a regular Java exception object. This automatic method for integration is applicable to the Eclipse IDE.

For BlueJ users, we provide a second distribution consisting of two JAR files. Adding both JAR files to BlueJ's existing lib directory is all that is required for installation. With this change, all student code run within BlueJ—whether it is invoked interactively through the mouse using the object bench, is initiated by code typed into the code pad, is a stand-alone program started from a class' main method, or is run as a JUnit test—will automatically receive ExceptionDoctor explanations on all uncaught exceptions.

6 An Evaluation of Effectiveness

Murphy et al. performed an analysis of the effectiveness of the messages produced by the ExceptionDoctor framework.[4] In their study, they tested 17 students' (8 male and 9 female) ability to fix a runtime error in provided source code. While solving the runtime exception, thirteen students were given ExceptionDoctor like output to aid their work. Of these students, 76% were able to find the cause of the exception within eight minutes. When asked if the messages were helpful, 13 (100%) students responded "yes". Two of the students who did not complete the task said that the exception message mislead them. However, they also admitted they had not closely read the message.

The four remaining students were given the same task, without the aid of ExceptionDoctor messages. These students were the highest performing members of the class and easily found the cause of the error within five minutes. Afterwards, when shown the ExceptionDoctor messages, three of the four students said that the ExceptionDoctor message would have helped them discover the cause of the error. One student felt the message would have not helped; she claimed that the message was too long (One of the reasons we reorganized the information presented to the students).

Our research results do not focus on the effectiveness of the messages produced by ExceptionDoctor since their effectiveness has been proven by Murphy et al.[4] We instead focus on the tools effectiveness in identifying exceptions that students normally encounter.

The code used as the basis for this evaluation consisted of all student program assignments submitted for a full semester of a CS2 course. Student work was collected using an automated grading system called Web-CAT [7]. Students were allowed to make unlimited submission attempts, and were required to write their own JUnit tests to demonstrate that their code behaved correctly. All submissions from all students—that is, both work in progress and final solutions—were used for this analysis.

While the majority of student code did not produce run-time errors, there were a total of 465 uncaught exceptions produced by tests that students wrote, and another 4,053 uncaught exceptions produced when instructor-written reference tests were run against the student solutions for grading purposes during the course.

Figure 9 shows the distribution of exception types produced by student-written tests. Of 465 uncaught exceptions, ExceptionDoctor provided useful explanations for 297 (64%). An additional 9 exceptions (2%) actually arose in library classes called by student code, rather than directly inside student code—in these cases, ExceptionDoctor pinpointed the location in the student code where the library call originated, but was unable to add information to the explanation because library source was unavailable. ExceptionDoctor failed to provide explanations for the remaining exceptions (34%) because it had no appropriate handler to provide an augmented explanation (other than the default handler for any Throwable). Of these, 136 (29%) were due to user-defined exceptions specific to a particular assignment. The remaining 23 exceptions (5%) do not show up as common for students in earlier studies. Additional handlers could be written if desired.

When student submissions were being more rigorously exercised by instructor-written tests, an order of magnitude more uncaught exceptions were produced. Of 4,053 exceptions, 3,882 were successfully explained (96%), and an additional 43 (1%) were due to exceptions occurring inside library code called by student code. Figure 8 shows the distribution of exception types encountered. Overall, ExceptionDoctor performed extremely well.

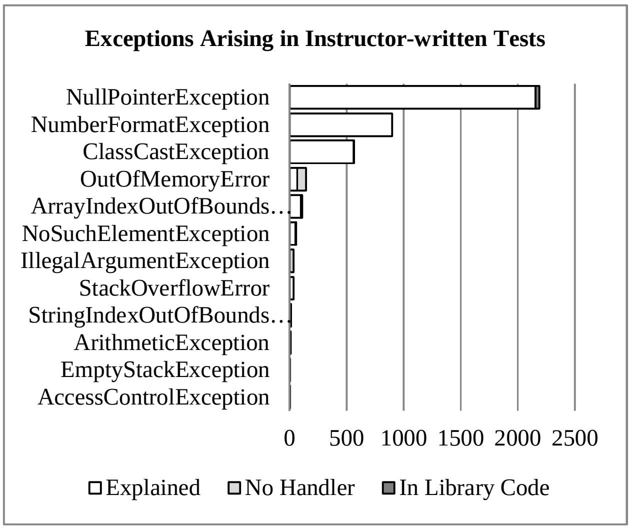


Figure 8: A graph of exceptions found and handled in tests written by instructors on code written by students.

These results are skewed towards exceptions that students could not debug themselves. Students also choose when to submit to Web-CAT for grading. Because they wrote their own tests and worked to make them pass, the exceptions occurring during grading have many of the simpler exceptions already removed. Further, problems that students could not resolve would show up repeatedly in multiple subsequent submissions, further skewing the frequency distribution.

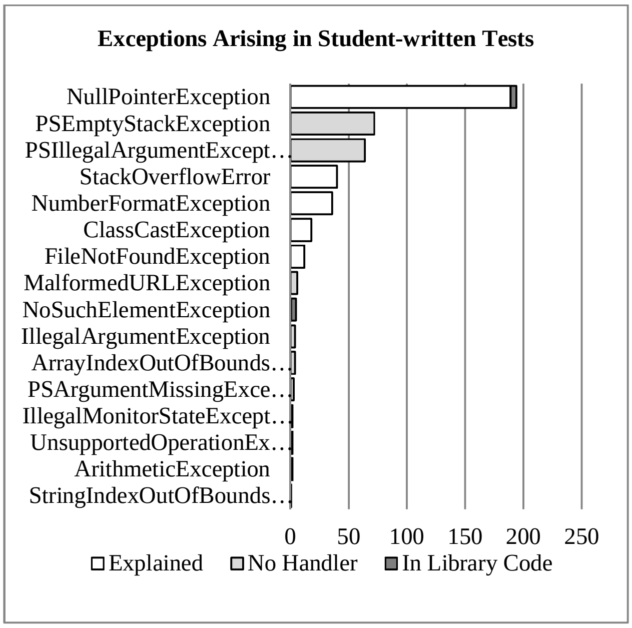


Figure 9: A graph of exceptions found and handled in tests written by students on their own code.

7 Future Improvement

ExceptionDoctor is easy to install with minimal overhead to use. In the current release, handlers are provided for exceptions that beginning programmers generally cause. However, the library of exception handlers could be improved over time to handle a broader group of exceptions.

The algorithms used by ExceptionDoctor could also be improved. The current version of ExceptionDoctor is limited to inspecting the single source line referenced by the original exception. To improve this, ExceptionDoctor could examine a larger portion of the source code looking for variable assignments or declarations.

ExceptionDoctor also struggles to identify exceptions thrown within library code. In these circumstances, the rewritten exceptions must contain messages indicating that a parameter to the library method resulted in an exception. In future research, we hope to design messages that teach students strategies for resolving exceptions they have caused indirectly.

ExceptionDoctor could also be improved through use of the Java runtime debugger. It might be possible to run the test concurrently in the Java debugger in an attempt to get real-time information about variable contents and execution information, rather than solely by examining the source code.

Further research needs to be conducted on student use of ExceptionDoctor. Plans have been made to integrate ExceptionDoctor into Web-CAT's Java grading plug-in so that all Web-CAT users can benefit from it. By logging exceptions that occur, more data on actual errors can be collected. Surveys could also gauge the helpfulness of exception message improvements in the eyes of students.

8 Acknowledgements

This work is supported in part by the National Science Foundation under Grant No. DUE-0618663. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9 REFERENCES

[1] JUnit home page, http://www.junit.org/

[2] Eclipse home page, http://www.eclipse.org/

[3] M. Kolling and J. Rosenberg, BlueJ home page, http://www.bluej.org/

[4] C. Murphy, E. Kim, G. Kaiser, and A. Cannon. Backstop: A tool for debugging runtime errors. In Proc. 39th SIGCSE Technical Symp. Comp. Sci. Education, pages 173—177, 2008.

[5] M. Jadud. A first look at novice compilation behavior using BlueJ. In Proc. 16th Workshop of the Psychology of Programming Interest Group, 2005.

[6] C. Norris, F. Barry, J. B. Fenwick Jr, K. Reid, and J. Rountree. ClockIt: Collecting quantitative data on how beginning software developers really work. In Proc. 13th Conf. Innovation and Technology in Computer Science Education (ITiCSE), 2008.

[7] Web-CAT home page, http://web-cat.org

Web-based System for Effective Student Management

Kwan Sun Choi, SaeRon Han, Jongsik Lim, Dongsik Kim, Dal Ahn , ChangWan Jeon, Sunheum Lee*, Sangyeon Woo**, HeungGu Jeon***

Department of Electrical and Communication Engineering,

*Department of Information Communication Engineering

SoonChunHyang University

**Department of Sports Science

SoonChunHyang University

***Department Electrical Engineering, Anyang University

E-mail: cks1329@sch.ac.kr, jeoncw@sch.ac.kr

ABSTRACT

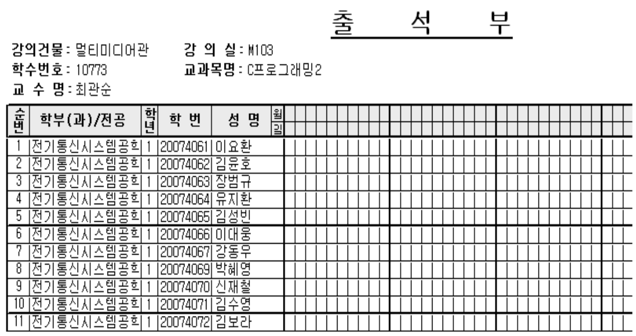
In this study student administration system based the .NET foundation was implemented that can effectively address the attendance processing and the student management tasks for numerous courses that constituents many students and professors. The system configuration is largely divided into student management server, student management client system, and attendance processing system. The attendance processing system performs automatic attendance processing by utilizing the student ID cards. Our system converts traditional roll books into database for more efficient management, and students can view the history of their attendance via the website and manage the records of their attending lectures conveniently.

Keywords: .NET, ASP.NET, ORACLE, LMS, Student Administration

1. INTRODUCTION

With the emergence of information technology and communication means particularly the computer and the Internet, the field of education is also taking the online form using them. Integrated with the Internet, software system required to accommodate changes to such a new paradigm is called LMS (learning management solution) as a collection of various information processing technologies. Each university is building LMS as well as developing diverse contents, and executing academic affairs such as information service, school administration process and E-Learning through the Web. In addition, they are making efforts to develop solutions for interaction between professors and students or among students using bulletin boards, Internet mails, etc.

Many lecturers have used paper attendance roll shown figure 1 at Soonchunhyang University. But it has not other information about a student except student's identification number, grade, name and department. It is very difficult for them to discriminate students by only the information. Thus they have experienced difficulty to administer students. In order to solve the problem, the paper attendance roll with student images shown figure 2 has been developed and used. Because it offers a lot of conveniences to check a student's attendance, many lecturers are using it widely.



출 석 부

강의건물 : 멀티미디어관 강 의 실 : M103
학수번호 : 10773 교과목명 : C프로그래밍2
교 수 명 : 최관순

순번	학부(과)/전공	학년	학 번	성 명
1	전기통신시스템공학	1	20074061	이요환
2	전기통신시스템공학	1	20074062	김윤호
3	전기통신시스템공학	1	20074063	장범규
4	전기통신시스템공학	1	20074064	유지환
5	전기통신시스템공학	1	20074065	김성빈
6	전기통신시스템공학	1	20074066	이대웅
7	전기통신시스템공학	1	20074067	강동우
8	전기통신시스템공학	1	20074069	박혜영
9	전기통신시스템공학	1	20074070	신재철
10	전기통신시스템공학	1	20074071	김수영
11	전기통신시스템공학	1	20074072	김보라

Figure 1. conventional paper attendance book

Figure 2. paper attendance book with student image

Thus along with the development of software tools, many program for student administration are developed and provide many lecturers with an efficiency of student administration. But in the process existing software systems are revealing their problems and limitations including the high cost of repair and maintenance, difficulties in managing transactions and data and frequent failures and, as a consequence, people are turning their eye to the development of .NET - based learning management software system. Moreover, due to its low cost of construction and maintenance, the system is drawing the attention of many universities hesitating about introducing existing software system.

The Microsoft .NET Framework is a software component that is a part of Microsoft Windows operating systems. It has a large library of pre-coded solutions to common program requirements, and manages the execution of programs written specifically for the framework. The pre-coded solutions that form the framework's Base Class Library cover a large range of programming needs in areas including: user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. The class library is used by programmers who combine it with their own code to produce applications.

Programs written for the .NET Framework execute in a software environment that manages the program's runtime requirements. This runtime environment, which is also a part of the .NET Framework, is known as the Common Language Runtime (CLR). The CLR provides the appearance of an application virtual machine, so that programmers need not consider the capabilities of the specific CPU that will execute the program. The CLR also provides other important services such as security mechanisms, memory management, and exception handling. The class library and the CLR together compose the .NET Framework.

2. Design and Implementation

2.1 Construction of software system

The software system developed in this study is largely composed of student management server system, student management client system, and attendance processing system. The server side is composed of database, Web server, and system management server, and the student management client system is composed of student management module for professors and lecture management module for students. In addition, the processing system is composed of unmanned attendance machine and attendance checking module. Figure 2.1 shows the general structure of the student management system.

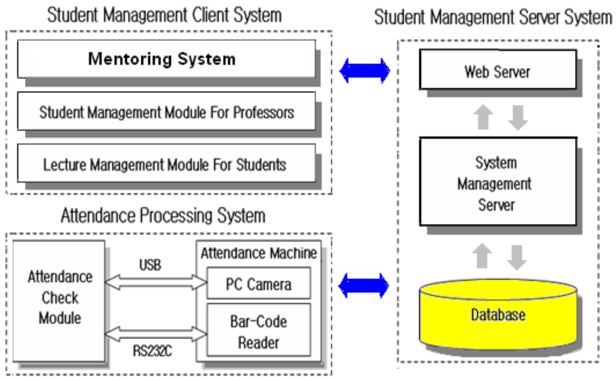


Figure 2.1 structure of the student management system.

2.2 Environment of Development

In order to develop student management server system, we have used environment as like table1.,2,3

Table 1. Developing Environment for student management server system

ITEM	SPEC.
CPU	Pentium4 3.0MHz
RAM	1024 Mbyte
OS	Windows 2003 Server Enterprise Edition
Database	Oracle8i Enterprise Edition 8.1.6.0.0
Web Server	Internet Information Server 6.0
Tool	Visual Studio .NET 2003

Table 2. Developing Environment for student management client system

ITEM	SPEC.
CPU	Pentium4 2.4MHz
RAM	512 Mbyte
OS	Windows XP Professional
Tool	Visual Studio .NET 2003

Table 3. Developing Environment for Attendance processing system

ITEM	SPEC.
CPU	Pentium4 1.8MHz
RAM	512 Mbyte
OS	Windows XP Professional
Tool	Visual Studio .NET 2003
PC Camera	AMTEKVISION ZECA MV 402
BARCode Reader	TEKSCAN TSK-750

2.3 Student management server system

The student management server system, which is composed of database, Web server and system management server, plays the role of a server for the entire student management system. The database was built with Oracle, and the Web server used IIS (Internet Information Server) and is compatible with development tool ASP.NET. In order to use Oracle, we used OLEDB(Object Linking and Embedding Database) Data Provider. The OLEDB such as data access object with COM based provides methods accessible to all kinds of data. The Connected database used and processed ADO.NET(ActiveX Data Object) functions in the module included the business logic layer. Figure 2.2 shows configuration of connection system and database

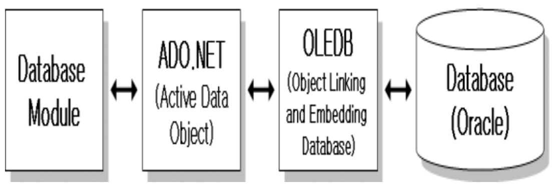


Figure 2.2 Configuration of database connectivity

The system management server is an application server in charge of the business logic layer that carries out major tasks of the student management system. In addition, it is the only part directly connected to the database in the software system, and all client modules in the student management system were designed to access the database through the system management server. The system management server is composed of student information management service, learning management service, and attendance processing system server. The server as a collection of functions carries out jobs such as student data processing, lecture - related tasks, attendance - related tasks, etc., and controls the general flow of the software system. Figure 2.3 shows the structure of the system management server.

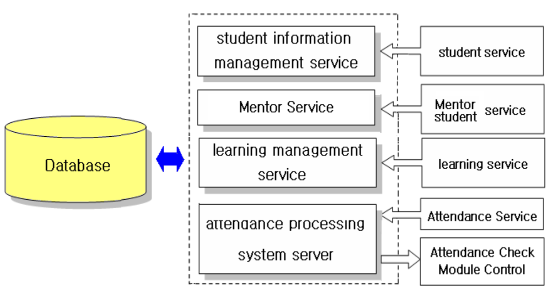


Figure 2.3 structure of the system management server

2.4 Student management client system

The student management client system is a client application containing modules for professors and students, providing functions such as student management and lecture management. The software system is composed of student management module for professors and lecture management module for students. The student management module for professors is an application developed using C# and ASP.NET, and the lecture management module for students is a website developed using ASP.NET.

2.4.1 Student management module for professors

The student management module for professors is an application for professors to manage students and lectures. It provides functions for the integrated and systematic management of data about lectures such as announcements, roll books and teaching materials as well as data about students attending the lectures. In addition, with the module, a professor can check students' attendance at his/her current lecture in real time, and can adjust the time of attendance checking.

Figure 2.4 and 2.5 are the screens of the implemented student management module for professors. The user can choose a lecture using the menu bar on the top and carry out tasks related to the lecture. Figure 2.4 shows the function of managing student information, and Figure 2.5 the function of setting the attendance checking time, which can set time for processing students' attendance, lateness and absence for the selected lecture.

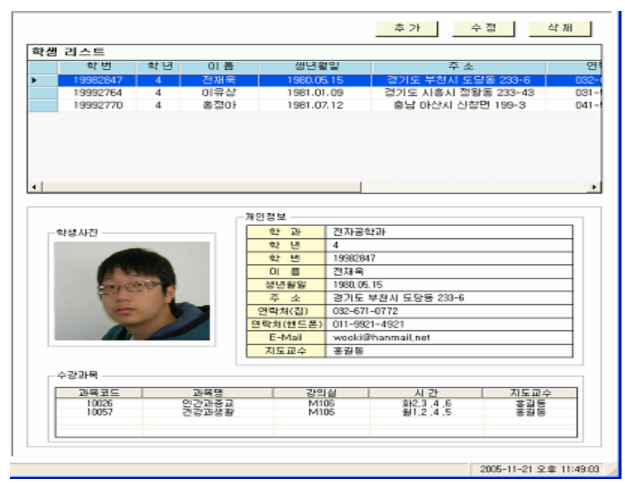


Figure 2.4 student management module for professors

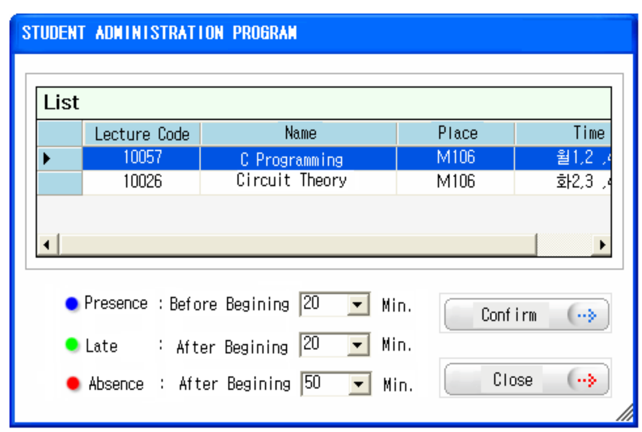


Figure 2.5 setting the attendance checking time

2.4.2 Lecture management module for students

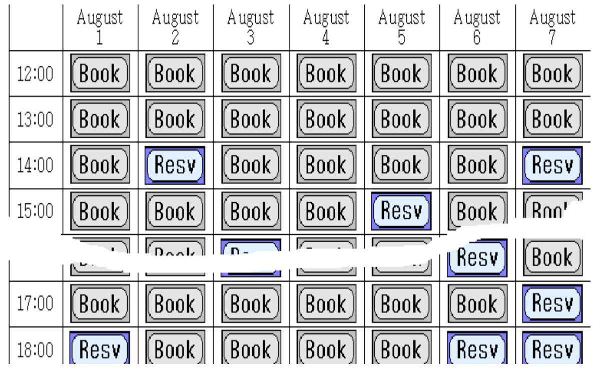
Students connect to the website and use services related to lectures that they are attending. Like the module for professors, this module also provides the function of viewing attendance. In addition, students can manage their attending lectures conveniently using announcements, the bulletin board for questions and answers and lecture materials. Moreover, because all these services are available just by connection to the Internet, the effect of lessons is improved.

Figure 2.6 is the screen of the implemented lecture management module for students, showing menus such as announcements and lecture materials. Figure 2.6 shows the function of viewing the history of attendance for the selected lecture.

Figure 2.6 Lecture management module for students

2.4.3 Mentor System

Our mentor system is a form of study guidance provided by student mentors during the first six months of the course. The mentor receives special training on guiding mentor groups. The faculties select the mentors on the basis of study progress and motivation. At the start of the semester, students are assigned to mentor groups. This is based on registration for first year C programming projects. The project group is therefore also the mentor group. Each group is given a student mentor (senior student or master students). The group normally consists of around 5 students. Over a number of sessions, the first year students receive information about the course and about student life, they can talk to fellow students and find out about study skills by booking time schedule. The booking timetable is figure 2.7. The aim of this system is to automate the booking procedure, to provide speed, flexibility and efficiency for students. In case of utilizing this booking system, they are not obliged to remain many hours to the department waiting for service.



	August 1	August 2	August 3	August 4	August 5	August 6	August 7
12:00	Book	Book	Book	Book	Book	Book	Book
13:00	Book	Book	Book	Book	Book	Book	Book
14:00	Book	Resv	Book	Book	Book	Book	Resv
15:00	Book	Book	Book	Book	Resv	Book	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Resv	Book
17:00	Book	Book	Book	Book	Book	Book	Resv
18:00	Resv	Book	Book	Book	Book	Resv	Resv

Figure 2.7 Booking table

2.5 Attendance processing system

The attendance processing system, which is deployed in each lecture room, was implemented for automatic attendance management using student ID card. The software system is composed of unmanned attendance machine and attendance check module. The unmanned attendance machine is for students to put in their student ID card, and the attendance check module controls the unmanned attendance machine and sends/receives data to/from the attendance processing system server

The Code 39 barcode is popular bar code symbology and can be read by almost all barcode scanners. The Code 39 is used for ID, inventory, and tracking purposes. There is code 39 in the student Identification card in figure 2.8.

The unmanned attendance machine is for students to put in their student ID card for attendance checking, and is composed a card slit and a camera. It reads a student ID card from barcode reader and sends the data through RS232C interface to the port connected the computer. Then the attendance check module receives and processes the data.

The camera in the unmanned attendance machine takes the images of students whose photograph has not been registered in the attendance checking database and stores them into the database automatically. Obtained images are transmitted from the attendance checking module to the attendance processing system server and saved in the database.

Figure 2.8 Student Identification card

3. Conclusion & Future work

In this study, we designed and implemented the .NET - based student management system that provides functions such as attendance management, student management and lecture management for professors and students.

We considered preferentially an efficiency of development due to reusable resources and a convenience of maintenance and repair in the processing of system design. We have developed various modules requiring each function. The student management system was constructed of them. We have constructed a component based distribution environment utilizing COM+(Component Object Model) and .NET Framework. Therefore our proposed system has a high expandability. To automate attendance management, we manufactured an attendance machine and proposed the methods of software system design and data processing. Through automating tasks related to attendance management, the implemented attendance management system provides convenience to both professors and students and enhances the efficiency of classes. What is more, it converts traditional roll books into database for more efficient management, and students can view the history of their attendance through the website and manage the records of their attending lectures conveniently.

We have surveyed program from lectures, students as like below.

1) It is helpful for lectures to administrate students effectively. (93% strong agree)
2) It is helpful to be familiar with students. (90% strong agree)
3) It is easy and convenient to use because of web-based program. (91% strong agree)

The response is 90% over at strong agree. The student management system was implemented in this study limits its scope to attendance checking, student management and lecture management. The mentor system of student management induced students to be familiar with mentor specially.

Thus, further research is necessary to implement a general learning management system and collaborative learning system that covers broad functions as well as its interoperation with various types of platforms including mobile devices. We are developing a student management system using RFID.

References

[1] Kwansun Choi, Wan-Seon Choi, "A Development of student management system for using PC camera" *thesis for a bachelor's degree, Univ. of Soonchunhyang*, 2002

[2] Don-Ho Kang, "A Design and Implementation of Student Management and Educational Guidance System based Web" *thesis for a bachelor's degree, Univ. of Kyungsung*, 2005

[3] Seung-Hoon Lim, "A Present State and Plan of System for Attendance Management" *thesis for a bachelor's degree, Univ. of Mokpo*, 2004

[4] Jae-Young Oh, "A Study on Learning Object based on COM", *thesis for a bachelor's degree, Univ. of Korea National Open*, 2004

[5] Chang-Hee Lee, "A Design and Implimentation of Education-Service System used ASP.NET" *thesis for a bachelor's degree, Univ. of Sejong*, 2003

[6] Hyang-Joo Lee, "Supplementary and Advanced Differentiated Teaching-Learning System on Web-based Instruction" *thesis for a bachelor's degree, Univ. of Hanseo*, 2003

[7] Kyung-Duk Son, "A Study on Imrpoving Performances of Distribute Applications based on .NET" *thesis for a bachelor's degree, Univ. of Kangwon*, pp. 91–95, 2005

[8] Dong-Wan Lee, "A Design and Implimentation of Project Management System based .NET Framework" *thesis for a bachelor's degree, Univ. of Konkuk*, 2003

[9] Oracle Web Page : http://technet.oracle.com

[10] MSDN (Microsoft Development Network) : http://msdn.microsoft.com

[11] SUN Java Web Page : http://java.sun.com

[12] Campione, Walrath, Java Tutorial Second Edition, Addison Wesley, 1998

[13] Kim Jongkeun, Oracle Bible ver.8.x, YoungJin Press, 1999

[14] ELMASRI/NAVATHE, Fundamentals of Database Systems, Addison Wesley

[15] http://en.wikipedia.org/wiki/Microsoft_.NET

[16] Vincent Chevrin. Project Ubi-Learn : an Intermediation Infrastructure for Multi-channel Accesses to Future LMS. AICT/ICIW 2006.

[17] Christian Floerkemeier. RFID middleware design - addressing application requirements and RFID constraints. Institute for Pervasive Computing Department of Computer Science. 2005.

[18] Robert C. Yoder, Using RFID in the Classroom to Teach Information Systems Principles. CCSC : Northeastern Conference, 2006.

Experiences Evaluating Student Attitudes in an Introductory Programming Course

Stephen H. Edwards[1], **Godmar V. Back**[1], **and Michael J. Woods**[1]
[1]Department of Computer Science, Virginia Tech, Blacksburg, VA, US

Abstract— *This experience report describes using a validated survey instrument to measure changes in student attitudes toward computing across a CS1 course. The choice of the survey instrument is described, with links to online resources for several existing instruments. The setup of a within-subjects design using pre- and post-tests at each end of a semester-long programming course allows changes in attitudes to be assessed. Results from analyzing 197 surveys received over a two-semester period are presented. Most attitude measures were quite high. The only significant change was an increase in the perceived importance of computing among the students. However, student aversion scores were a significant predictor of student success. Lessons learned are presented along with future plans.*

Keywords: Perceptions, Computer Attitudes Survey, CS1

1. Introduction

As at many other institutions, our faculty are constantly striving to recruit a broader population of students, encourage those with an affinity for computer science to stay in the discipline, and offer them learning opportunities that students find satisfying. This is particularly true in our introductory courses where attrition is the highest. While a student's decision to major in computer science depends on a number of factors, his or her freshman year—including their first college-level class in computer science (CS1)—plays a crucial role [13].

At the same time, however, it is important to employ appropriate strategies for measuring progress and assessing effectiveness. This experience report describes the methods used and results obtained when trying to assess the impact of curricular changes on student attitudes toward computing in the CS1 course at Virginia Tech.

Recently, our department introduced a new approach to the programming activities and assignments in our CS1 course in order to appeal to a broader, more diverse student audience. Rather than using canned problems from mainstream textbooks, we looked to several recurring themes to increase the appeal of computer science [12], [6], [3]: using **real-world problems** that are relevant in a student's life; using problems that show the **social and societal impact** of our discipline; creating solutions that are **meaningful to others** outside the classroom; employing **modern interaction techniques** (such as web-based GUIs and graphical output); and giving students more **creative control** and more flexibility in the solutions they create.

The underlying rationale overarching these themes can be stated simply: Students are more interested in solving problems that they find personally relevant, so looking for ways to increase the relevance of assignments across multiple dimensions is important for attracting a broader and more diverse population. We chose to employ new educational materials that allow the use of state-of-the-art Web 2.0 technology in CS1, enabling a new generation of programming activities that combine all of the themes described above. Our hope was to improve the recruitment and retention of computer science students, and to give students in our CS1 course a more positive view of our discipline and its impact.

As part of our plan to evaluate the impact of these changes, we were interested in whether or not there was a significant (positive) change in the attitudes our students hold toward the field of computing. In addition to exploring retention data, course success rates, and changes in student demographics, exploring possible changes in their attitudes will allow us to investigate whether our curricular changes might be giving students a more positive view of our discipline.

This paper reports our initial experiences in gathering and analyzing student attitudinal measures during two semesters of the CS1 course at Virginia Tech. Section 2 outlines CloudSpace, the Web 2.0 infrastructure that makes web-based programming assignments easy in CS1. Related work on instruments to measure attitudes toward computing are discussed in Section 3. Our study design is presented in Section 4, and results are presented in Section 5. The paper concludes with lessons learned over the first two semesters of this evaluation task.

2. Background: CloudSpace

As described in Section 1, the modifications to our CS1 course focus on linking computer science education to real-world contexts that illustrate the social and societal impact of computer science. We also believe that students should be given creative control to produce personally relevant applications that also are usable by others in a meaningful way. Emerging technologies that underlie new Web 2.0 applications and the sleek interaction techniques they enable create a unique opportunity to provide such relevant context for the teaching of computer science concepts.

Capitalizing on this idea is difficult, however, because most tools for producing Web 2.0 applications are aimed at professional programmers and are not "student friendly." They presume one has already mastered programming, and are not level-appropriate for use in an introductory course like CS1.

To address this shortcoming, we designed and built CloudSpace [14], a web development platform that brings Web 2.0 application production down to the level where CS1 students can take advantage of modern techniques while still learning the basics of programming. Using CloudSpace, students can write plain old Java code, and then write a user interface using XHTML with a few custom tags. Student Java code then executes on the CloudSpace server, and the student's XHTML is automatically transformed into an interactive, direct-manipulation-based, AJAX-style web interface viewable in most modern web browsers.

Through the CloudSpace server, all of the support infrastructure necessary to take advantage of modern Web 2.0 features is completely encapsulated and hidden from students. The mechanics of the interface layer are completely transparent, so that students need only learn Java (on the programming side) and HTML plus a few custom tags (on the user interface side).

CloudSpace runs every student's individual web application in a separate virtual machine, providing full access to basic Java features such as file system access, console output, and exception diagnostics, just as one would have when running Java code locally from the command line. Student code can be run locally and even unit tested in the same way as other plain Java code, and then "skinned" with a modern web-based interface by adding a separate XHTML file. This approach minimizes the number of additional concepts students must learn in order to develop web applications. Beyond regular CS1-level Java programming, they primary addition is learning XHTML and some basic CSS styling—concepts the students are already eager to engage, in many cases.

With CloudSpace, virtually any conventional Java assignment can be given a web interface. For example, a simple body mass index calculator can be given a web-based facelift, deployed as an interactive web page that dynamically updates its results as different numbers are typed into text boxes. This has the added benefit of deploying the program in a way so that the student can show it off to anyone, including family, friends, or classmates in other courses.

But CloudSpace really shines when it is used to create assignments that speak more directly to the web-based world where students spend time every day. For our CS1 course, we have developed a number of assignments centered on building and extending the features common to many social networking sites, such as Facebook. We were originally inspired by an assignment presented at the Nifty Assignments special session at SIGCSE 2008, where Mehran Sahami presented a Facebook-inspired assignment from Stanford University. With CloudSpace, students can build live social networking applications with dynamic AJAX-based features easily, using simple CS1-level Java programming.

Giving students activities with this style of programming is important because these web-based applications increasingly determine the public face of computer science because of their ubiquity and their large number of users. We have been using this approach to programming activities successfully in CS1 at Virginia Tech for two years. However, the question of whether these types of assignments affect student attitudes toward the discipline of computing remains, driving the evaluation activities described in the remainder of this paper.

3. Related Work

There is a wide body of research literature on alternative mechanisms for measuring one's attitudes toward computing that goes back decades. Most such research has traditionally been performed in the field of education rather than computer science. However, much of this research is valuable for CS education researchers and is readily available for application in CS courses.

Most often, attitudes towards computing are measured using self-reported answers on a survey instrument. Many such instruments exist; Christensen and Knezek provide an overview of many [1]. Of these, The Computer Attitude Scale (CAS) developed by Loyd and Gressard [10] is one of the most widely used. In addition to providing a measure of general attitudes toward computers, the CAS also provides four subscales measuring one's *anxiety* when dealing with computers, *confidence* in dealing with computers, *liking* or enjoyment of dealing with computers, and perception of *usefulness* or proliferation of computers on future jobs.

The CAS has been used by other CS educators to investigate student attitudes as an indicator for success in a CS1 course. In study at Saint Joseph's University, researchers studied the use of Alice [11], a 3D programming environment, to teach students with a high risk of failure in a CS1 course the basics of software development. In the study, students were grouped into three categories: high risk students who took the Alice course, high risk students that did not take the Alice course, and low risk students that did not take the Alice course. Students were scored using grades, retention to the computer science department, and attitude surveys. Their evaluation results showed that the Alice course offering was an effective educational tool for rasing student's grades and retaining these high risk students in the computer science department. However, use of Alice did not result in any significant changes in attitude measures on the CAS subscales between pre- and post-test values.

Farkas and Murthy also applied the CAS to measure changes in attitudes across a computer literacy course with an introductory programming component [4]. Students in the course were given the CAS survey at four evenly spaced points ranging from the beginning of the semester to the end of the semester. The study results indicated a statistically significant decrease in positive attitudes toward computing during the first two-thirds of the course, when programming activities were present. The authors concluded that the course did not “seem to be successful in creating a positive attitude towards computers.”

Other CS education researchers have sometimes developed their own instruments. Stuart Hansen and Erica Eddy [5] developed a short survey for use with the computer science projects at the University of Wisconsin to evaluate the level of engagement and frustration experienced by students. The researchers theorized that a higher level of engagement would correlate with a low amount of frustration. Instead, they found that engaging projects did not result in lower levels of frustration. On further inspection of their results, they did see a correlation between engaging projects and the student’s attitude towards the project’s subject matter.

In education, other instruments are used in some situations. Among them, one is particularly interesting. Knezek and Christensen developed the Teachers Attitudes Towards Computing questionnaire (TAC) by aggregating all or part of a number of other survey instruments with strong track records [1], [2]. The TAC included all of the questions in the CAS instrument. They were particularly concerned with drawing on the best existing work, using validity and internal reliability measures to identify the best-performing instruments. With an initial version containing 284 questions and 32 separately identifiable subscales, the TAC was experimentally assessed for validity[2]. The analysis showed that the 284 question survey actually measured between 4-22 subscales out of 32. Over time, the TAC has been successively refined to remove redundancy and retain subscales with the highest reliability. After reducing the TAC down to 16 subscales, and then ten, recent research is focusing on the seven-factor version of the questionnaire. These factors include enthusiasm, anxiety, avoidance, email, negative impact on society, teacher productivity, and semantic perception of computers. Since its creation, the TAC has been used in multiple studies around the world.[8], [7], [9].

For other CS education researchers, the research web site hosted at the Texas Center for Educational Technology provides electronic access to several independently validated survey instruments for measuring attitudes toward computing: http://www.tcet.unt.edu/research/. Complete copies of several versions of the TAC, the Computer Attitude Questionnaire, and several related instruments are available, including full question sets, survey descriptions, lists of references, and scoring procedures.

4. Method

While using the CAS to measure student attitudes is common, we were interested in exploring other potential factors, including perceptions of the field as a viable future vocation. In exploring alternative survey instruments, however, the TAC was particularly attractive, since its developers had focused on subsuming and then refining the best of available instruments into a better composite. However, the TAC did pose three problems. First, some versions of the TAC are quite long, which can be detrimental to survey response rates. Second, the TAC includes some subscales aimed specifically at teachers (e.g., attitudes toward teacher productivity, use of e-mail in classroom management, etc.). Third, as the TAC as evolved and the number of subscales (and questions) has been reduced, many subscales appropriate for students (e.g., attitudes toward computing as a future career) have been dropped as being of less interest when surveying teachers.

As a result, we chose to employ a customized version of the TAC that includes subscales we are most interested in, omits subscales specific to teachers, and keeps overall survey length to a practical size. The resulting instrument, which we can call "TAC-lite", includes the entire CAS question set plus six additional subscales. Giving this survey provides the following twelve measures:

- *CAS Anxiety* indicates one’s anxiety in dealing with computers.
- *CAS Confidence* indicates one’s confidence in dealing with computers.
- *CAS Liking* indicates one’s degree of enjoyment when dealing with computers.
- *CAS Usefulness* indicates one’s perception of the utility of computers and their prevelance in future occupations.
- *CAS Overall Score* is a combination of the four subscales and indicates one’s general attitudes toward computers overall.
- *Importance* indicates how important you perceive computers to be to in your future occupation or work life.
- *Anxiety* is the TAC subscale indicating one’s anxiety towards computers.
- *Acceptance* indicates one’s acceptance of (or, alternatively, desire to avoid) using computers.
- *Vocation* indicates one’s perception of computing as a future occupation area.
- *Aversion* indicates one’s negative opinions about people in the computing field.
- *Relevance* indicates one’s perception of computing as a relevant topic for study and educational pursuits.

All questions included in this TAC-lite survey were Likert-scale questions. Except for the Importance subscale, all questions were answered on a 5-point scale ranging from Strongly Disagree to Strongly Agree. The Importance subscale questions used a 4-point scale without a neu-

Table 1: Summary of Measures for Fall 2009/Spring 2010

Measure	Pre-Test	Post-Test
CAS Anxiety	4.36	4.48
CAS Confidence	4.31	4.32
CAS Liking	4.25	4.29
CAS Usefulness	4.48	4.53
CAS Overall Score	174.02	176.76
Importance	4.29*	4.43*
Anxiety	4.35	4.41
Acceptance	4.66	4.67
Vocation	4.39	4.45
Aversion	4.12	4.00
Relevance	4.37	4.45

tral/undecided value. The scoring procedure for the TAC was used for all subscales.

After narrowing the TAC to these specific subscales, the corresponding questions were published along with basic demographic questions in an on-line survey. During the fall semester 2009 and spring semester 2010, all students in our CS1 course were given an opportunity to complete the on-line survey once at the beginning of the semester, and again at the end of the semester. A total of four sections of the course were offered each semester, consisting of approximately 200 students in total. The survey instrument is available electronically at http://cloudspace.cs.vt.edu/computer-attitude-survey/.

Participation in the survey was optional for all students. Since our CS1 course is a lab course and lab attendance is mandatory, the opportunity to complete the survey was included as an optional part of the lab activity during the second week and again during the final week. Collecting both pre- and post-test answers to the survey allows a within-subjects comparison to look for changes in each individual's self-reported attitude data.

5. Results and Discussion

Response rates for the survey were approximately 50%, with slightly more students electing to complete the survey at the beginning of the course, and slightly fewer electing to complete it at the end of the course. A total of 197 completed surveys were received. All subscales except for the CAS Overall Score produce results on a 1-5 scale where 1 is most negative/undesirable and 5 is most positive/desirable. In particular, "negative" subscales, such as Anxiety and Aversion, follow this same pattern (e.g., an Anxiety score close to 5 indicates low anxiety, while a score close to 1 indicates much stronger anxiety). The CAS Overall Score gives results on a 40-200 point scale, where 40 is most negative and 200 is most positive. Table 1 summarizes the scores on the pre- and post-tests.

The most notable aspect of this result is that both the pre-test and post-test attitudes are very positive towards computing compared to most other published studies. This is likely due to the population. About three quarters of

Table 2: Summary of Course Scores by Aversion Level

Aversion Level	Proportion of Students	Mean Score
>= 4.93	6.7%	89%
>= 3.33	84.0%	70%
< 3.33	9.3%	63%

the students enrolled in our CS1 course are CS majors or prospective majors, and a substantial portion of the remainder are intent on or considering a CS minor. As a result, the incoming population already has definite opinions about the field. It is reassuring to see that their views are so positive. In many other studies, the subject pool is much broader (literacy courses, high school students, introductory courses taken for general education credit by students from many majors, etc.).

To examine changes from pre-test responses to post-test responses, a within-subjects model was chosen so that each student's post-test answers would be compared against their corresponding pre-test results. We performed an analysis of variance on each subscale using the semester and the pre- or post timing as independent factors. From the point of view of the changes made to our assignments, however, only one indicator shows a significant change between pre- and post-test values at the $p = 0.05$ level. The Importance subscale shows a small but significant increase in post-test values. The standard deviation for this subscale was 0.51, so the net increase of 0.14 is only 27% of one standard deviation. While statistically significant, the size of this shift is minor.

Since our students responded extremely positively on both pre-tests and post-tests, it appears that they already had very positive attitudes toward computing on all of the instrument's measures coming into the course. Fortunately, the results do not indicate that our programming activities contributed to a more negative perception of computing, as in Farkas and Murthy's study [4].

Because the responses on all scales were so positive, we also investigated the smaller proportion of students who responded somewhat less positively on the survey. After eliminating all students in the top half of respondents, we repeated the statistical test to see if the lower half of respondents showed any significant changes from pre-test answers to post-test answers. The results were the same, suggesting that trends in the lower half and upper half of the respondents did not diverge.

In addition to examining changes between the pre- and post-test, we also examined whether any of the pre-test scores were useful as predictors of success in the class. Student responses on the pre-test were matched with student final scores in the course. Then a stepwise regression was performed with the score as the dependent variable and the entire set of subscales as the potential factors.

Of all the factors, only one had a statistically significant predictive relationship with final course score: the Aversion subscale. Understandably, students who rated themselves as being more averse—that is, holding more negative opinions about people in the computing field—were more likely to score lower. Students who rated themselves as less averse were more likely to score higher. Applying an optimal partitioning algorithm revealed three distinct clusters to the students. An upper threshold of 4.93 (out of 5) on the Aversion subscale (indicating someone who has little aversion and a very positive attitude) characterized the top 6.7% of respondents on this subscale. Those students averaged final cumulative scores of 89% (an A- or B+ grade). A lower threshold of 3.33 (out of 5) on the Aversion subscale (indicating someone who is undecided or who has a more negative attitude toward people in the computing field) characterized the bottom 9.3% of respondents on this subscale. Those students averaged final cumulative scores of 63% (an F grade, or a possible course drop/withdrawal).

6. Lessons Learned and Conclusions

We have only begun the process of evaluating the impact of the changes made in this CS1 course, and changes in student attitudes are only one outcome. Since this is a common issue facing many CS educators, however, this experience can be useful in outlining a possible path for others. It has also revealed a number of potential lessons.

One important suggestion for other CS educators is to consider the set of existing instruments when undertaking an attitude survey. Using an instrument that has been published and that has been checked for validity and internal reliability is very helpful, when many researchers cannot afford the time and effort it takes to perform this foundational research on their own custom instruments. There are multiple instruments to choose from, and using a common instrument may allow you to consider other published work as a basis for comparison.

It is also important to consider your subject pool. In our case, the incoming students already had very positive attitudes toward computing. It may not be practical to set a goal of increasing their attitudes significantly, since there is so little room for improvement relative to the standard deviations of the measures. However, you will never know where your population truly lies until you measure. In this respect, the effort that we put into the attitude surveys has revealed that students (well, those who elect to complete the final survey) leave the course with very positive attitudes, which is consistent with our goals. Also, there is no evidence that student attitudes weaken (among those who completed the final survey), which is an important concern.

One question this raises, however, is related to attrition. What about students who choose to leave the course (drop the course, withdraw, or just give up and stop attending)? In practice, those students are unlikely to complete the final survey, since they likely have left the class before the last week. This necessarily limits our post-test data to the (voluntary) responses of students who stayed in the course. A more pro-active mechanism using some kind of "exit survey" for those who drop or withdraw would be an important addition. After all, it is those students who are most likely to have negative changes in their attitudes, but they are also to most likely to skip the final survey.

This evaluation project is a work in progress, and this experience report only describes the results so far based on the first two semesters of survey data. In the future, a more comprehensive analysis that includes retention results (which students progress to the next course, data on dropped/withdrawn students, separate predictor analyses for tests and for programming activities, etc.) will extend this work. As other CS education researchers approach similar problems, however, sharing experiences is the best way to learn from each other's successes (and speed bumps).

References

[1] R. Christensen and G. Knezek. Internal consistency reliabilities for 14 computer attitude scales. *Journal of Technology and Teacher Education*, 8(4):327–336, 2000.

[2] R. W. Christensen and G. A. Knezek. Construct validity for the teachers' attitudes toward computers questionnaire. *Journal of Computing in Teacher Education*, 25(4):143–155, 2009.

[3] J. P. Cohoon. An introductory course format for promoting diversity and retention. In *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pages 395–399, New York, NY, USA, 2007. ACM.

[4] D. Farkas and N. Murthy. Attitudes toward computers, the introductory course and recruiting new majors. In *Group, Sussex University*, pages 268–277, 2005.

[5] S. Hansen and E. Eddy. Engagement and frustration in programming projects. In *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pages 271–275, New York, NY, USA, 2007. ACM.

[6] D. Hendrix, D. Umphress, and J. H. Cross. Designing a first-year project course to engage freshman software engineers: An experience report. In *Software Engineering Education and Training, 2006. Proceedings. 19th Conference on*, pages 25–34, 2006.

[7] G. Knezek, R. Christensen, E. L. Gilmore, H. Kim, A. D. Magoun, C. Owens, C. Morales, B. Moonen, and J. Voogt. Teacher and student attitudes toward information technology in four nations. In J. Price, J. Willis, D. A. Willis, M. Jost, and S. Boger-Mehall, editors, *Proceedings of Society for Information Technology and Teacher Education International Conference 1999*, pages 916–918, Chesapeake, VA, 1999. AACE.

[8] G. A. Knezek and R. W. Christensen. Attitudes toward information technology at two parochial schools in north texas. Technical Report 97.2, Texas Center for Educational Technology, 1997.

[9] G. A. Knezek and R. W. Christensen. *Instruments for Assessing Educator Progress in Technology Integration*, page 19. 2000 Institute for the Integration of Technology into Teaching and Learning, 2000.

[10] B. Loyd and C. Gressard. Reliability and factoral validity of computer attitude scales. *Educational and Psychological Measurement*, 44(2):501–505, 1984.

[11] B. Moskal, D. Lurie, and S. Cooper. Evaluating the effectiveness of a new instructional approach. In *SIGCSE '04: Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pages 75–79, New York, NY, USA, 2004. ACM.

[12] L. Rich, H. Perry, and M. Guzdial. A CS1 course designed to address interests of women. In *SIGCSE '04: Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pages 190–194, New York, NY, USA, 2004. ACM.

[13] E. Seymour. *Talking About Leaving: Why Undergraduates Leave the Sciences*. Westview Press, 2000.

[14] M. Woods, G. Back, and S. Edwards. An infrastructure for teaching CS1 in the cloud. In *ASEE Southeast Section Annual Conference*, 2010.

A FRAMEWORK FOR TEACHING COMPUTATION TO STEM STUDENTS IN E-LEARNING ENVIRONMENT (TECSSEE)

Chi Ben
Department of Computer and Information Sciences
Florida A&M University
Tallahassee, FL 32304
xbenchi@gmail.com

Hongmei Chi
Department of Computer and Information Sciences
Florida A&M University
Tallahassee, FL 32304
hchi@cis.famu.edu

ABSTRACT— To address the lack of computer assisted learning environments designed specifically for education in STEM (science, technology, engineering and mathematics), we propose a web-based e-learning environment via visualization, integrated with social learning, to meet the need from different disciplines in STEM (TECSSEE). The web service comprises of a three tier organization, including demos, hands-on labs, and quizzes. In the demo, students are shown how to use a certain application; during hands-on labs, students are asked to solve real-world problems in an active learning style, which means the students need to manipulate all the input data, parameters and rules, and see how data is visualized according to different factors, to learn about the concepts; last but not least, quizzes are given to the students as examinations of the learning. In addition, social learning will be incorporated in this system, students are provided with a social networking site offering group discussion, group blogs and wikis as further assistance to each specific hands-on lab.

Keywords: Visualization, e-learning, blended learning, hands-on labs, U learning, active learning, social learning.

1. INTRODUCTION

STEM education refers to the fields collectively considered core technological underpinnings of undergraduate teaching and learning. Many STEM classes incorporate computer assisted teaching methods to some extent. Virtual teaching platforms, such as Blackboard, have been used in a variety of teaching institutions. Blackboard develops and licenses software applications and related services to over 2200 education institutions in more than 60 countries (Blackboard Inc.). In Spain, 71.8% of university teaching and research staff make use of institutional virtual teaching platforms and 92.5% of the students make use of such platforms (CRUE, 2009).

The current trend to complement traditional face-to-face classes with electronically supported teaching and learning is known as “e-learning”; this style of learning is usually used to reference out-of-classroom and in-class-room education via technology. Under situations where student numbers are high, the use of e-learning can produce better outcomes in learning practices and patterns; they are widely applied in large-scale classes, such as first year of undergraduate study, which is an important year to help students build up commitments to university learning (Huon, 2007).

E-learning has gone through two phases, from simply reconstructing textbook contents which are taught in the classroom to building up various delivery models for the students. Numerous approaches have been designed to meet the need, the teaching media include, but not limited to, technologies such as Internet, intranet/extranet, audio or video tape, satellite TV, and CD-ROM.

Currently, widely used e-learning systems comprise of mainly general purpose applications, such as Blackboard, GOAL and Moodle. They are good extensions of traditional classroom activities, such as lecture, homework, and tests. An e-learning platform like this is not fully utilizing the power of computer and network, only limited capability has been offered. In this paper we will introduce a framework of e-learning where students can visualize course content and interact with the platform in an active learning pattern so that better teaching and learning efficiency is achieved.

2. RELATED WORK

Hazem M. El-Bakry, et al, have proposed Service Oriented Architecture (SOA) as a design pattern that presents systems as collection of reusable services that can be exposed and consumed on the Internet with standard interfaces (Hazem M. El-Bakry, 2009). Maja ukušić, et al, designed and assessed a comprehensive model for managing the e-learning process and to define the relationship between systematic implementation of the model, outcomes of certain e-learning aspects and subject of e-learning (Maja ukušić, 2010). Chao Boon Teo and his colleague have pointed out that current LMS are still limited to just being online repositories, lacking of learning personalization, and they have proposed a methodology for eliciting and personalizing tacit knowledge (Chao Boon Teo, 2006).

Chaofan Wang, et al, introduce visualization techniques to the teaching and learning process of the highway engineering experiment (Chaofan Wang, 2010). An E-portfolio is introduced to the Distance Chinese Language Visualization Teaching Platform, then designs the function modules of E-portfolio according to the function requirements analysis is described, and the application of E-portfolio in Chinese learning and the evaluation process based on it is explained (Yi Zhang, 2008).

3. THE E-LEARNING WEBSITE

This e-learning environment is focused on providing visualization experiences for STEM education, including biology, chemistry, computer science, mathematics, and etc., as shown in figure 1. All the disciplines are independent. Each contains one or more hands-on labs providing insights to a certain project where students are required to work by themselves. On this site, an interactive visualization style is used throughout all visualizing labs, which means the system will interact with students' input. As one input a set of data, the system will present visualized figures. If input data is changed or modified, the output will be adjusted accordingly.

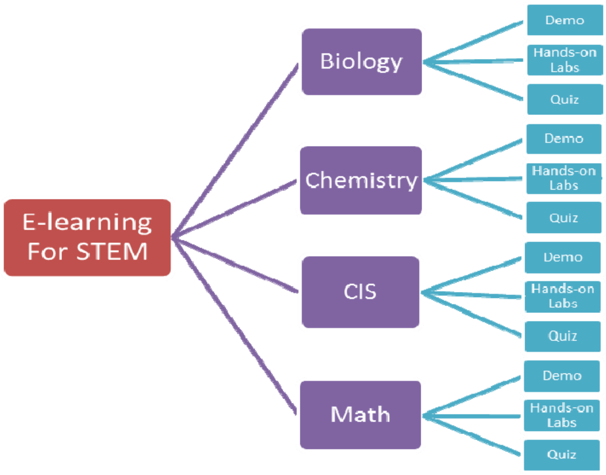


Fig. 1 3-tier of TECSSEE

The website will not be open to public, for the reason that we are currently focusing on providing STEM disciplines within Florida A&M University only. As seen in figure 2, each student will have a profile stored on the website, at the time of login, authentication is required. Student profiles are to be created by administrators.

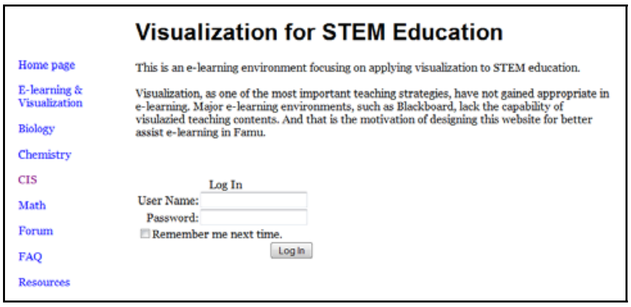


Fig. 2 Homepage with logon

The navigation menu on the left provides accessibility to all the resources on this website, including entries to lab projects of STEM disciplines, and other useful information. The main part of this site will be constructed in Microsoft .Net Framework; however, other script languages may be incorporated as well when necessary.

4. DESIGN OF TECSSEE

The goal in this paper is to create a visualizing environment for STEM education, different from other approaches, including extension of traditional classroom interactions such as Blackboard, advanced teaching and research software packages such as Matlab which is complicated and costly to general students, and simple applications with very limited functionality, such as the Visualization Tools for Teaching Computer Security website (Visualization Tools for Teaching Computer Security), which provides static demonstrations of information security issues without offering real hands-on labs to the students. Our site will be employing visualization in an active learning style with real hands-on labs so students have "learning by doing" experience.

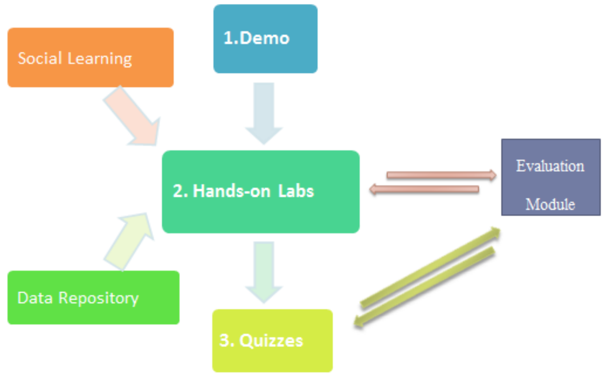


Fig. 3 Structure of TECSSEE

Such experience will be offered in a three tier style assisted by social learning and a data repository, as shown in figure 3.

4.1 DEMONSTRATION

One of the most significant problems with current e-learning applications is that the learning curve is sharp. When getting to a new environment, students will often experience frustration when there are no simple and easy guidelines to follow. To address this issue, we will use Jing screencasting computer program and open source video recording software CamStudio to record how each application works, and offer students the recorded video or images so they can learn about the system in a timely manner.

4.2 HANDS-ON LABS

The hands-on labs are where students practice and improve their knowledge during solving real-world problems. By designing the visualization experience according to course schedule, it will provide students good supplement to textbooks.

Figure 4 shows the screenshot of a visualization lab which reconstruct social networks using their datasets, with appropriate algorithms, to reveal important social networks patterns. By selecting data type in the list on the left, and an algorithm on the right list, the system is able to display the social networking graph; in this case, community detection is used, then after clicking the "Generate" button, the graph with identified communities within it has been displayed.

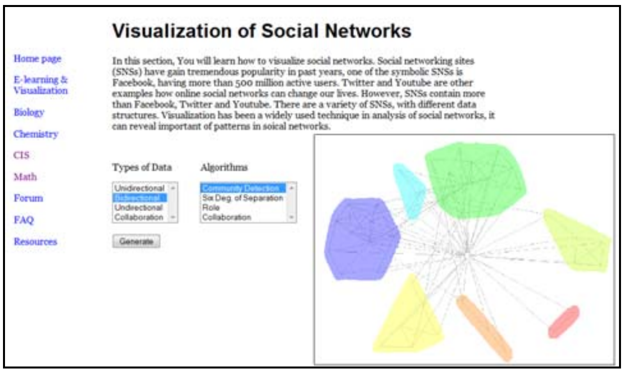


Fig. 4 Prototype of social network visualization page

Another case study is web-implemented Maxima. Maxima is an open source computer algebra system, which is usually installed on computers, either on Windows, Mac OS or Linux.

Our Computation for STEM Education project has involved Maxima as one of the teaching tools. During the use of this software, we have got response from the students about a variety of issue other than using the system. Installation is one of the most common problems. To solve it, we will implement Maxima on the website, so with a modern browser, anyone can take advantage of this powerful system without having to worry about installation.

As shown in figure 5, the web-implemented computer algebra system contains a main window, which takes input to it and then gives output, including graphs. By providing examples of the commands to students, and let them try the commands, it will not has a very sharp learning curve. The simple interface will not make students feel overwhelmed. But as one discovers more of the system, he or she will realize it can accomplish complicated work.

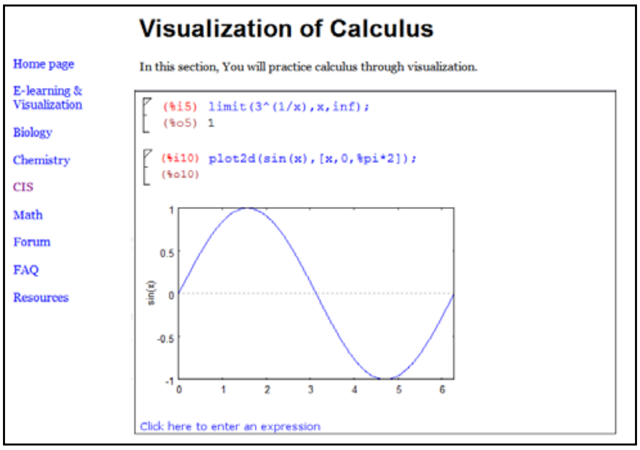


Fig. 5 Conceptual screen of a math algebra system based on Maxima

4.3 QUIZZES

Quizzes will incorporate visualization as well as general questions. Taking into consideration that general questions such as multiple choices or fill in the blank will conform to grading standards, they will be preferred types of questions. After taking quizzes, the results are evaluated and discover the weakness of students. The hands-on labs will be given for students to enhance the concepts until students master those concepts via hands-on labs.

4.4 SOCIAL LEARNING

One of the most important features of e-learning 2.0 is social learning, which assumes that the best way to learn something is to teach it to the others. The prosperity of social networking sites has met this demand. Open source social networking software has made it possible to create social networking sites with only a little effort.

The most important reason about creating a social learning networking site is that it is dedicated to the learning experience provided by the visualization site. Although tremendous commercial social networking sites are available, and almost all of them provide group discussion and other features, students will be reluctant to discuss academic problems within their social circles. A dedicated network will provide the convenience for their academic /professional needs. In such a virtual environment, students will be relaxed and ask or answer any questions while they can review repository of related questions and answers from previous semesters. In addition, such sites will be easily access from their social networks such as Facebook, MySpace.

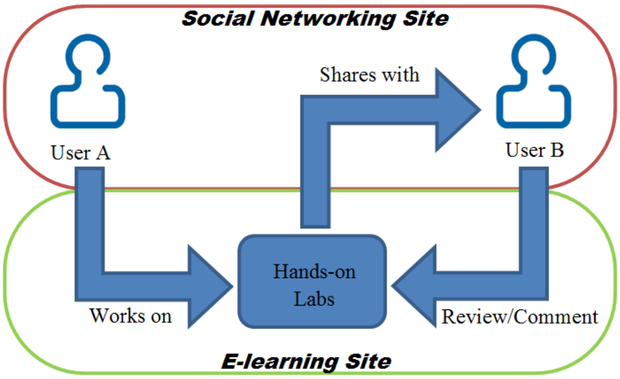


Fig. 6 Collaboration using the social learning features

Figure 6 demonstrates how a supportive social networking site can serve as a communicative platform for our e-learning framework. One of the great benefits from web 2.0 technology is users are allowed to join a social network and share all kinds of information among them. In this case, user A has doubts about a hands-on lab and asks user B for help by sharing current work with user B. After looking at the work, user B can review the work by giving comments. User B may be another current student, a previous student, or an instructor.

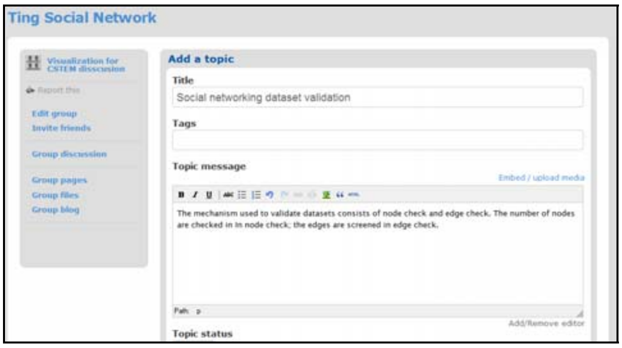


Fig. 7 Posting a discussion on the social networking site

As shown in figure 7, the social networking site has been set up using elgg social networking software. It provides group discussion and group blog pages. Each group can create their own blogs that are shared among all group members, but not to the public. See figure 8, in the access control a user is able to specify who will be able to view this blog post. This group blog setting ensures privacy and promotes efficiency.

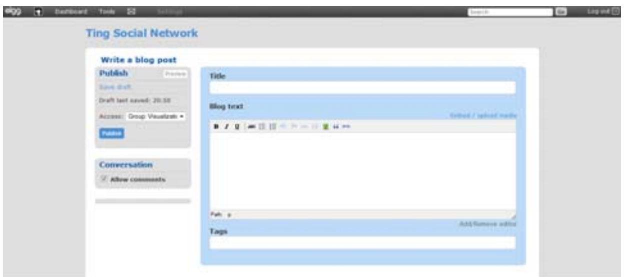


Fig. 8 Screenshot of posting a group blog in the Ting social network

4.5 DATA REPOSITORY

Visualizing experience requires data, for instance, to see how stock market goes, one needs historical stock data. A data repository will be built, which includes all the necessary data, such as social networking datasets, and biology genome datasets. Many types of data can be obtained from a variety of organizations, making real world based visualization possible.

5. CONCLUSION AND FUTURE WORK

Most current learning management systems fail to provide interactive visualization to the students, mere extensions of classroom activities will soon bore the students. Complicated systems are often research oriented; students will be overwhelmed by their interfaces, if the teaching material is not well-designed. This paper presents a novel web-based teaching environment, with the power of visualization and active learning. We do not target on providing tons of functions, just to offer what is really necessary for the course schedule.

Future work will include the completion of the visualization website, and adding various social forums to the social learning networking sites.

ACKNOWLEDGMENTS

This work has been supported in part by U.S. Department of Education grant P120A080094 and P120A090122.

REFERENCES

Learning management system. (2011). Retrieved 2011, from Wikipedia: http://en.wikipedia.org/wiki/Learning_management_systems

Blackboard Inc. (n.d.). Retrieved from Wikipedia.org: http://en.wikipedia.org/wiki/Blackboard_Inc.

Chao Boon Teo, R. K. (2006). A knowledge-driven model to personalize e-learning. *Journal on Educational Resources in Computing (JERIC).*

Chaofan Wang, Z. S. (2010). Teaching Highway Engineering Experiments by Visualization. *WMSVM '10 Proceedings of the 2010 Second International Conference on Modeling, Simulation and Visualization Methods* (pp. 267-270). Washington, DC, USA: IEEE Computer Society.

CRUE. (2009). *Conferencia de Rectores de las Universidades Españolas (CRUE).* Madrid.

Hazem M. El-Bakry, N. M. (2009). Advanced technology for E-learning development. *CIS'09 Proceedings of the international conference on Computational and information science 2009* . Stevens Point, Wisconsin, USA : World Scientific and Engineering Academy and Society (WSEAS).

Huon, G. S. (2007). Resource use and academic performance among first year psychology students. *Higher Education*, 53, 1-27.

Jeffrey Heer, d. b. (2005). Vizster: Visualizing Online Social Networks. *IEEE Information Visualization (InfoVis)*, 32-39.

Maja ukušić, N. A. (2010). e-Learning process management and the e-learning performance: Results of a European empirical study. *Computers & Education*, 554-565.

Visualization Tools for Teaching Computer Security. (n.d.). Retrieved from Information Security: http://www.ncat.edu/~xhyuan/security_visual_tools/

Yi Zhang, Q. L. (2008). Application of E-Portfolio in the Distance Chinese Language Visualization Teaching Platform. *ICWL '08 Proceedings of the 2008 Seventh International Conference on Web-based Learning* (pp. 76-81). Washington, DC, USA: IEEE Computer Society.

Teaching Ethics of The Cyber World

Suhair H. Amer, Ph.D.
Department of Computer Science, Southeast Missouri State University, Cape Girardeau, MO, USA

Abstract—*Students from all academic disciples use cyber-technologies, yet many don't realize the ethical issues associated with such use. In the author's institution a cyber-technology ethics course was developed and the author shares her experience of designing, implementing and offering this course.*

Keywords: Cyber-ethics, computer ethics, technology ethics, cyber-technology ethics

1 Introduction

Cyber-technology refers to a broad spectrum of computing and information technologies that range from stand-alone computers to the cluster of networked computing, information, and communications technologies. In this course, the moral, legal and social issues and controversies involving cyber-technology will be discussed. The broader social and ethical concerns, such as privacy, security, crime, intellectual property, and Internet regulations that affect each of us in our day-to-day lives are also covered. Questions about the roles and responsibilities of computer and information-technology professionals in developing safe and reliable computer systems are, also, examined. In this course students will examine cyber-world ethics examples and scenarios to prepare them to apply such theories in real-life situations. Students will also apply and extend existing ethical theories to the cyber-world.

2 Interdisciplinary Nature of the course

Students from all academic disciplines use cyber-technologies. They use personal computers, laptops, and handheld devices, and they access the internet. They interact with friends and family through MySpace, Facebook, and Twitter, and they view and upload videos to YouTube. They interact with other people in virtual worlds and participate in multiplayer online games. They complete assignments using Microsoft Word and reference Wikipedia as an encyclopedia. Cyber-technologies, whether emerging or converging, are implanted in many personal items, home appliances, cars, and public transportation. Using these technologies raises some ethical, social, and professional challenges that affect each of us in our day-to-day lives [1].

Issues and controversies that comprise cyber-ethics will be examined. Actual and hypothetical case studies will be discussed where students can bring their own discipline and personal experiences. All students will find many issues covered in this course pertinent to their personal and professional lives. Students will verbally and nonverbally communicate their ideas and opinions regarding cyber-technology-related ethical issues. They can use evidence, reasoning, and example to elaborate upon their opinions. In this course, a research paper is required and the students will present a summary of their research findings. Students will engage in discussions, and debate and advocate their opinions. In addition, written assignments will assess students' ability to relate the material covered in class to their own experiences and case studies in their world. Students will apply discussed and debated cyber-world ethics to real-life situations. They will also relate many cyber-technology ethical issues to variations of existing ethical problems.

3 Purpose or Objectives of the Course

Students when taking this course will become familiar with some organizations, laws, and regulations related to a Professional Code of Ethics as well as ethical concepts and theories of different cultures, disciplines, and agencies. They will learn how to critically analyze contemporary cyber-related ethical issues and assess ethical aspects of emerging and converging technologies. In several class sessions students will engage in discussions penetrating the social impact and ethical issues that emerge from the widespread use of cyber-technology. Finally, they will apply the knowledge acquired in personal and professional decision–making situations related to cyber-technology.

4 Expectations of Students and Basis for Student Evaluation

There will be weekly assignments mainly consisting of short essay questions which assess the students' comprehension of key concepts, themes, issues, and scenarios covered in the course.

There will be weekly discussions designed to encourage students to reflect more deeply on some of the controversial issues examined in the course.

There will be one research paper and one presentation. Students will select one controversial ethical topic related to the use of cyber-technology. In their paper, students will define the technology, discuss some of its uses, mention the advantages and disadvantages of using the technology, discuss controversial ethical issues related to its use, and finally state their conclusion. Finally, each student will present a summary of his/her findings to the rest of the class.

The exams will consist of multiple-choice questions, True/False questions, matching questions and short essay questions. Exams are used to assess students' comprehension of facts, concepts, theories and topics discussed in class.

5 Course Outline

While developing this course, the author investigated several attempts to teach similar computer or technology ethics courses. The author found that [1] and [2] outlines most of the important topics. In general, the course is divided into three major components. First, the student is introduced to the history of cyber technology ethics. This is achieved by first laying out the foundation of technology and information ethics. Cyber technology is then defined and the basics concepts and theories are introduced. In addition, it is important to introduce students to the different Codes of Ethics that may apply to their major of study and future careers. The second component deals with critical thinking for evaluating cyber-ethical issues. In this component, students will acquire the skills to reason and provide a systematically sound argument regarding different issues. Students will enhance their critical thinking skills, understand logical arguments, and understand how to apply the tools for evaluating cyber-ethical issues. This will allow students to establish and justify a moral system. Finally the third component discusses basic concepts and ideas in cyber-technology ethics, such as: professional and moral responsibility, privacy and anonymity, computer system security, crimes involving technology, intellectual property and ethical behavior, commerce and speech and responsible conduct, globalization, ethical aspects of emerging technologies.

6 Design, Implementation and Assessment Considerations

Blooms taxonomy [3] consists of cognitive, affective, and psychomotor domains. It concentrated mainly on the cognitive (mental) domain. Anderson and Krathwohl [4] revisited Bloom's taxonomy to refocus attention and assist educators with problems associated with the design, implementation and assessment of curriculums. Similar to Bloom's Taxonomy, student's learning progresses from memorization of facts to application of concepts in a distinct functional domain. However, Anderson and Krathwohl expands the taxonomy from one dimensional to two dimensional consisting of categories of factual knowledge, conceptual knowledge, procedural knowledge and meta-cognitive knowledge. The cognitive category consists of several processes such as remembering of facts, understanding concepts, applying concepts, analyzing problems, evaluation of related concepts, to finally applying concepts in a distinct function domain.

While designing the course, the different levels of the Anderson and Krathwohl's [4] cognitive model were considered. Since in a higher level course, students go beyond remembering and understanding different concepts. They are required to apply covered theories and concepts to current case studies and situations. Analyzing and evaluating decisions is also a major component of the course.

In the general, the following objectives were considered while developing the course:

6.1 Demonstrate the ability to locate and gather information

Each student will select an ethically controversial topic related to the use of cyber-technology. For example, the instructor found the following topics appropriate: biometrics[5], virtual reality[6][7], intelligent user interfaces[8], and nanotechnology[9]. A paper and a presentation are due towards the end of the semester. Students select the resources that match their topic and that hold valuable information related to their topic. In addition, some assignments require students to locate examples of case studies supporting their views.

In the beginning of the semester, the instructor will briefly mention the different topics that will be covered in the course and ask students to select an ethically controversial topic related to the use of cyber-technology. The research paper and presentation are student-centered activities. The instructor will provide feedback throughout the process.

To evaluate the student's ability to locate and gather information, he/she will be required to finish different requirements leading to the research paper and presentation such as 1) Identify a research topic, 2) Gather information and select recent refereed articles and/or books as references, 3) Prepare an outline, 4) Write a first draft of the paper, 5) Modify first draft and submit a final draft, and 6) Summarize findings and prepare a presentation. Each stage is assigned a deadline and a grade to ensure that students will follow these deadlines.

In addition, some assignments require students to locate additional resources to become familiar with organizations, laws, regulations, and ethical concepts of different cultures, disciplines, and agencies related to cyber-ethics.

6.2 Demonstrate capabilities for critical thinking, reasoning and analyzing.

The course covers several ethical topics involving cyber-technology. As each topic is covered, essential theoretical concepts are explained, and case studies are discussed. The students analyze the roles of the victim and the attacker. They will also study different ethical theories discussing moral/non-moral issues resulting from the use of technology.

Several cyber-technology-related ethical topics are covered. The instructor will initiate activities such as encouraging students to read related material, then discussing and analyzing additional case studies. The instructor will discuss and explain the process of logical reasoning and how to provide a systematically sound argument. The instructor, in discussions, will guide and make sure that students present their ideas and opinions using a systematically sound process. In addition, the student will be directed to apply an ethical method to reason and isolate its different components. Sometimes, they will be asked to provide feedback on why they think some actions should have been performed and what they think would prevent such actions from being repeated.

6.3 Demonstrate effective communication skills.

Writing and discussion will be used as vehicles to communicate students' ideas and opinions. Students will express, formulate, and apply the concepts covered in the course. This will be addressed through discussion, written assignments, a research paper, and an oral presentation.

Through the different assigned activities, the instructor aims to improve the quality of written performance by giving feedback. For the stages leading to the final draft of the research paper, the instructor will provide feedback and require revision. If the student needs additional help, he/she will be encouraged to attend writing tutorials offered at the University Writing Lab. Guidelines on how to participate effectively in discussions will be provided to the student at the beginning of the semester. As the semester progresses, students will become more familiar with the process and be more comfortable with participating. Students who prefer not to participate in discussions will be encouraged to prepare a one-to-two-minute talk about a topic related to the discussion. All students will be required to prepare a presentation, due towards the end of the semester, regarding the research papers.

Written assignments, discussions, and the different stages leading to the research paper and presentation are spread across the semester. The students should show evidence of applying studied concepts to provide valid arguments.

This course addresses the needs of any student who uses a computer, the internet, smart phone, or any type of cyber-technology. This applies to the breadth and diversity of the entire student body experience. For example, the students in the course will become aware of the fact that when people participate in social networking services, they are unwittingly giving up more of their personal privacy via Web-mining.

Although new technologies emerge and existing technologies evolve, many of the ethical issues associated with them are basically variations of existing ethical issues. However, many emerging technologies present us with challenges that do not fit easily into our conventional ethical categories[1]. In the course we will discuss how those controversies can be analyzed from the perspective of standard ethical concepts and theories. In addition, since personal computers, laptops, cell phones, etc. play an important part of a student's life, it is important to explain what ethical issues relate to such usage. Students are also encouraged to discuss their personal, professional, cultural, and educational experiences with regard to using cyber-technologies.

6.4 Demonstrate the ability to make informed, intelligent value decisions.

The aim of the course is to place value on the individual's decision-making. The students are required to draw, when possible, an ethical, moral, and practical conclusion. When dealing with cyber-technology, such decisions involve well-being. Moral implications of activities are discussed and studied in the course, and students are required to make informed and intelligent decisions based on the different code of ethics and the case studies discussed.

In discussions and assignments, students should clearly state their opinion and draw from the content discussed in class as well as general documented knowledge from their own disciplines and experience. For example, the student should demonstrate that he/ she is able to relate case studies and a code of ethics to give a clear, well-thought-out decision or opinion. The student should demonstrate satisfactory performance on the exams, which demonstrates that he/she read the assigned material.

7 Discussion and Conclusion

Students from all academic disciplines use cyber-technologies. They, for example, use personal laptops, handheld devices, access the internet, and use Facebook and YouTube. They interact with other people in virtual worlds and participate in multiplayer online games. In addition, they use many educational applications to complete their assigned work. The author realized the increasing importance of developing a cyber-technology related ethical course for students in different majors in the authors' institution. The paper discusses the components and her considerations while developing the cyber-technology ethics course.

In general, the students indicated the following to be some of the strengths of the course:

- Students enjoyed taking the course and they enjoyed the topics covered throughout the semester.
- In addition to exploring the case studies explored in the content of the course, students brought their own experience and examples from their own fields of study.
- Diversity of cyber-related topics and diversity of students' fields of study made the course more interesting.
- Students appreciated that they were given a chance to choose from among several cyber-technology-ethical related topics for their research paper and presentation.
- Some students were surprised from some of ethical issues and case studies discussed in class.

Some students also indicated some concerns such as:

- Few students are still not comfortable in writing a research paper.
- Due to some student's working schedules, they were unable sometimes to read and complete assigned activities and finish them by due dates.
- Few asked for more interactive activities.
- Students had to develop their discussion skills. As the semester progressed, they were able to enhance their critical and analytical skills and use them in their discussions.

In conclusion, the instructor believes in the importance of understanding the ethical issues related to the use of cyber-technology. It is critical to develop and maintain the course in a manner that is beneficial to students from all fields of study. It should be specific and concentrate on cyber-related-technologies ethical issues and general enough to incorporate case studies from all academic programs.

8 References

[1] Herman T. Tavani. Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing. 3rd edition. Wiley 2010.

[2] Terrell Ward Bynum and Simon Rogerson, Eds., Computer Ethics and Professional Responsibility, Blackwell, 2004.

[3] Bloom, B.S. (Ed.) (1956) Taxonomy of Educational Objectives, the classification of

educational goals - Handbook I: Cognitive Domain New York: McKay.

[4] Anderson, L. W. & Krathwohl, D. R., Airasian, P.W., Cruikshank, K. A., Mayer, R. E., Pintrich, P.R., Raths, J., and Wittrock, M. C. (Eds.) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon. Boston, MA: Pearson Education Group.

[5] Alterman, Anton. A Piece of Yourself': Ethical Issues in Biometric Identification. Ethics and Information Technology 5.3 (2006): 139-50.

[6] Hoffman, Hunter G., Patterson, David R. PhD, Seibel, Eric PhD, Soltani, Maryam Med, Jewett-Leahy, Laura BA, and Sharar, Sam R. MD. "Virtual Reality Pain Control During Burn Wound Debridement in the Hydrotank". Clinical Journal of Pain. 2008 May; 24 (4): 299-304.

[7] Darics, E. (2010). Politeness in computer-mediated discourse of a virtual team. Journal of Politeness Research: Language, Behavior, Culture, 6(1), 129-150.

[8] Wærn, Annika. "What Is an Intelligent Interface?" SICS. Web. 09 Feb. 2011.

[9] Cynthia Selin. Expectations and the Emergence of Nanotechnology. Science, Technology, & Human Values Vol. 32, No. 2 (Mar., 2007), pp. 196-220.

Binary Blaster: An Educational Game for Practicing Binary Number Conversions

Irene Polycarpou[1], Julie Krause[1], and Monica Noring[1]
[1]Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO, USA

Abstract - *Understanding of numbering systems, especially binary numbers, is of great importance in computer science and engineering education, but students have difficulties understanding such concepts. In addition, students' acquisition of new knowledge and skills may be enhanced by varying the presentation of instruction and by providing engaging activities, such as educational computer games, in which students build or practice those skills. This paper discusses an engaging educational game called Binary Blaster which has been designed for students to practice number conversion between decimal and binary numbers, as well as the results of a small exploratory survey on Binary Blaster with undergraduate computer science students.*

Keywords: undergraduate computer science and engineering education; binary numbers; number systems; conversion of binary numbers; conversion of decimal numbers; educational games.

1 Introduction

Among the most fundamental concepts in computer science and engineering education are numbering systems [11]. There are many ways to express numerical values using symbols, some of which are familiar and used everyday, and others of which seem foreign and awkward at best. Positional numbering systems have been in use since the 5th century A.D. [18]. Using positional numbering systems greatly simplified arithmetic over non-positional numbering systems, and the decimal system (or to be more specific, base-10 positional notation) became the most commonly used numbering system in the world.

With the rapid advancements in technology and access to it, one could argue that the binary system, or base-2 positional notation, is even more widely used than the decimal system. In fields such as computer science and engineering understanding of binary numbers is essential. All data on a computer, including the computer's instructions to be executed, are represented at their most basic level with binary numbers (without reducing those numbers further to simple high and low voltages). Professional organizations such as IEEE and ACM, identify numeric data representation as a core topic for a computing curriculum (e.g., IEEE Computer Society and Association for Computing Machinery Joint Task Force on Computing Curricula) [3]. Since data in computers are represented in binary form, an understanding of the binary number system is critical to understanding computer data representation. Furthermore, specific representations such as two's complement and floating point build on and require an understanding of binary numbers [7].

Most computer science and engineering undergraduate programs introduce the concept of binary numbers and techniques used for converting between binary representations and other number-base representations such as decimal and hexadecimal early in their curriculum (e.g., intro to CS courses). The IEEE/ACM Joint Task Force on Computing Curricula lists converting numerical data from one format to another as a learning objective for teaching machine-level representation of data. Any computer science or engineering student needs to have a basic understanding of the binary number system before s/he can ever fully understand the basic operations of the machine. Even more so, a student must understand the process behind binary arithmetic and operations performed by computers before s/he can develop useful and efficient software or hardware.

Learning the binary numbering system can be a tedious process [7]. There is an inherent difficulty in teaching and learning such an unfamiliar concept. It is generally accepted that counting and doing arithmetic in base-10 comes as second nature to humans. Using a differently based system is unnatural [1] and difficult for the human mind to comprehend [6]. Petzold (2000) describes conversions between the common number systems as tricky and awkward [4]. Moreover, without practice representing numbers outside of the decimal system, it may be difficult to become proficient with representing numbers in binary form. According to Bjork & Druckman (1994), "real competence comes only with extensive practice" (as cited in [9], p. 273).

The use of educational electronic games is one way that has been shown to be effective for improving learning and if used as a "coinstructional strategy" along with other pedagogical methods can enhance classroom instruction [5]. With the advancement of technology and the increasing number of electronic games targeting students, it is not a surprise that that many researchers argue educational games' potential to support education [2, 8, 12, 16], especially in terms of motivating, exciting, and engaging the students in the learning process. Computer games have been shown to be engaging and effective learning tools [5, 10]. Furthermore, educational games can capitalize on students' enthusiasm about electronic games in general and serve as a motivation for their learning as well as a medium for capturing students' interest in a specific subject area.

In this paper, we discuss an electronic educational game called Binary Blaster, which is designed for students to practice number conversions. Binary Blaster capitalizes on games' potential to help students learn in an immersive environment. Through the game, students are required to convert numbers from decimal to binary form under timed conditions.

2 Binary Blaster

Common categories of computer games include adventure games, puzzle games, role-playing games, strategy games, sports games and first-person shooter games [5]. Binary Blaster is a puzzle-type game designed to be used for practicing and gaining proficiency with number conversions (decimal to binary numbers). The game is intended to offer an immersive way of completing multiple choice problems under timed settings. The idea of the game is similar to the "tetris" game that most students are familiar with [17]. During game play, students are presented with a number in decimal form. This number is visually embedded in a colored shape such as a star or circle and it is initially placed at the top of the screen and gradually falls to the bottom of the screen (see Figure 1). At the same time, a variety of distinct binary numbers (with a uniform number of bits), including the binary number corresponding to the given decimal number, are displayed at the bottom of the screen. Students need to match the given "falling" decimal number to the correct binary number at the bottom of the screen, so that they can earn points. The form of the numbers displayed at the bottom of the screen can be either in standard binary representation, excess notation, or two's complement. Depending on what students want to practice with, they can select one of the three representations to start the game.

To add a difficulty level component to the game the speed of the falling number increases as the player progresses through the game and earns more points (i.e., identifies the correct number conversions). Users can also change the difficulty level of the game by selecting the number of bits in the binary numbers provided. Game difficulty levels are discussed in more detail later in this section.

Mayo (2007) recommends awarding points to improve a player's sense of self-efficacy, or believing in one's ability, as a means of improving learning outcomes through electronic games [10]. The player's goal in Binary Blaster is to earn points and achieve a high score. This goal is intended to provide motivation for students to play for extended periods, thus increasing time spent on task.

Currently, there are two modes available in the game, the single-player mode and the multi-player mode.

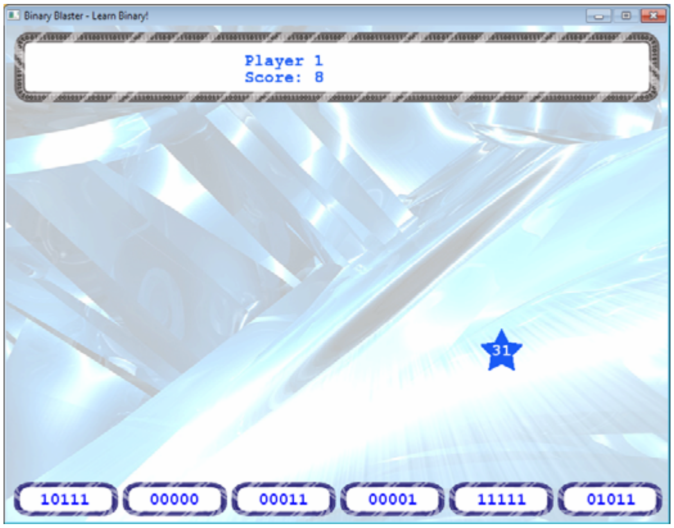


Figure 1: A screenshot from the single-player mode of Binary Blaster showing a decimal number shape (blue star) "falling" down the screen, along with the available binary representations for the "falling" decimal number.

2.1 Single-player Mode

In single-player mode, only one student is playing the game and the student can play the game until a desired score is obtained or until the student feels comfortable with number conversions. When the game begins, the score is set to zero. As mentioned earlier, the user is given a decimal number embedded in a shape ("falling number") and while that shape slowly falls to the bottom of the screen, the user should select one of the binary numbers available at the bottom of the screen and align the shape to it (see Figure 1). Currently, there are six numbers available for the user to choose from. Users can move the shape side-to-side, using designated left/right key presses, until the shape aligns vertically with the correct binary number. Once the shape reaches the bottom of the screen, the game responds according to whether or not the binary number selected was the correct one. As the shape falls, if the user wishes to accelerate the shape's journey to the bottom, the shape can be moved directly to the bottom using a designated down key. If the player chooses the correctly matched binary number, the score is increased by one and the successful match is visually displayed with a "correct" message and a dynamic "blast" display (see Figure 2). Alternately, if the binary number does not match the decimal number, the score is reduced by one (or remains the same if the score is currently zero) and an "incorrect" message is displayed. Correct and incorrect answers are accompanied by positive and negative sounds, respectively. After the user is provided with feedback on his/her response, a new number is placed within the user's shape and the shape once again begins to fall from the top of the screen. The entire "number-falling" process repeats until the user has reached a desired score and/or has become comfortable with converting decimal numbers to their binary form.

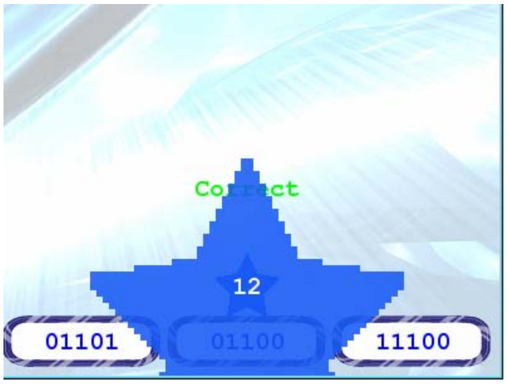


Figure 2: A screenshot from Binary Blaster showing a successful match of the given decimal number to a corresponding binary number (visual "blast").

2.2 Multi-player Mode

In multi-player mode several students can play the game simultaneously and compete with each other for the highest score (see Figure 3). Each player is provided with a different decimal number to be converted which is embedded in a unique colored shape for each player. The winner of the game is the player with the highest score after a predetermine period of time.

Figure 3: A screenshot from the multi-player mode of Binary Blaster showing three different players, represented by three differently colored shapes.

Each player has its own score that is updated independently of other players (see Figure 3). Also, players' speeds adjust independently based on their own scores. The game is designed such that when multiple users are playing, no player's involvement with the game is interrupted by other players' actions. For example, the list of binary numbers from which to choose is only updated when a player gets an answer correct, so as not to disrupt other players' goals. That is, if one player's target is a 7 (0111 in 4-bit binary), the 0111 option at the bottom of the screen will not "disappear" when another player mistakenly selects 0111. Moreover, in multi-player mode, the color used for a given player is consistent between the printed score and the falling shape and to avoid any confusion the colors used for each of the different players are distinct from one another.

2.3 Difficulty Levels

The game is intended to be immersive. As stated in Oblinger (2006), "games and play may be effective learning environments, not because they are 'fun' but because they are immersive…" [5, p.2]. Binary Blaster continually increases the level of challenge. The game starts with number shapes falling slowly, but the speed for a given player increases as s/he becomes more proficient, as indicated by an increasing score. When the player's score reaches five or a subsequent multiple of five (10 points, 15 points, etc.), the speed of the falling number increases, providing an additional level of challenge for the player. Mechanisms are built in to prevent the speed from impeding a player's progress as well. Before the speed has reached a point at which the player's decimal value is difficult to perceive, the speed level is capped. Moreover, the speed of a player's falling number is reduced if s/he appears to be struggling with the conversions (i.e., when the score drops below a multiple of five, the speed of the falling number drops).

In addition to the speed of the falling numbers, there are other factors in the game that change the level of difficulty. These difficulty levels are selected by the user(s) prior to the beginning of the game. One difficulty level is the number of bits in the binary numbers displayed at the bottom of the screen. The range of possible bit string lengths is three to eight. Three-bit numbers (representing eight distinct values) provide a sufficient challenge for beginners but may allow for memorization of patterns. Eight-bit numbers (representing 256 distinct values) provide a requirement that more mental calculations be completed.

Another difficulty factor is the number of targets (binary numbers to choose from) displayed at the bottom of the screen. The default level of the game is six targets, but more targets could provide additional challenge in that they would require the user to visually scan additional numbers to locate the correct one.

Finally, as mentioned previously, the game offers conversions between decimal numbers and three different representations of binary (standard binary, excess notation, and two's complement). While there may not be any representation that is more difficult to master than another, two's complement and excess notation often require additional calculations when converting from decimal form. Also, those representations include negative numbers. As a result, there is potential for those two representations to be more difficult, or require more time, than standard binary numbers.

2.4 Implementation

Binary Blaster was implemented using Python, version 2.6, and Pygame, version 1.9.1. The game runs on computers with Linux, Windows or Mac operating systems. Python [14] and Pygame [13] need to be installed on the user's computer so that the game can run. These downloads are free and the installation process typically takes no longer than 5 minutes.

Users can control the game via keyboard inputs and as an alternative, video game controllers such as Wii remote controllers can be used.

Currently, Binary Blaster includes conversions between decimal and binary representations such as standard binary, excess notation, and two's complement, but it is easily expandable to include other numbering systems such as hexadecimal (base 16 positional system) and octal (base 8 positional system).

2.5 Educational Usage

Binary Blaster is structured to be used as a supplemental instructional method in introductory Computer Science and Engineering courses at the high school or university level. Instructors can use the game as part of classroom instruction for students to practice and master number conversion (individual or group classroom activity). Students can also use the game on their own time and their own pace to practice number conversion or prepare for a test. It is expected that instructors using the game will provide instruction on binary representation prior to introducing the game.

All the numbers provided in the game are generated randomly, removing the limit on the number of problems students practice with. Students have the choice to run and re-run the game until they are satisfied with the number of problems they solved and feel that they mastered number conversion.

The game also provides immediate feedback to the students on their responses, as opposed to waiting for feedback from submitted assignments. This will allow students to learn from their mistakes. Actively learning, making mistakes, and learning from them, is more effective than passively listening to a lecture [15].

Since students can use the game outside the classroom this provides an additional convenience for them and allows them to learn according to their own needs. Instead of learning, studying, and practicing the material only during class time or through a textbook, students can use the game to practice on their own time, without having to forgo the benefit of feedback and without being "penalized" for incorrect responses (e.g., submitting homework with incorrect responses). Allowing students to work in a comfortable environment at their own pace and be in control of the learning process can motivate them to use the game.

3 Evaluation: Exploratory Survey

Before continuing expansion of Binary Blaster and incorporating it into the classroom, we decided to acquire feedback from undergraduate computer science students on whether Binary Blaster is something they would like to use as part of their learning of number conversions and whether they enjoy playing the game. Participating students were eight undergraduate students taking the Introduction to Computer Science course during the Spring 2011 semester. As part of the course, students were introduced to numbering systems and specifically, to the binary number system. We organized "playing sessions" with participating students after formal instruction in the classroom. During the playing sessions students were able to play Binary Blaster for 10 minutes (approximately, five minutes were spent on 4 bit and five minutes on 5 bit binary numbers). Most students only practiced with numbers in the standard binary form; two students used excess notation and/or two's complement. At the end of their session students were asked to fill out a questionnaire about Binary Blaster, so as to provide us with feedback on their experience. The questionnaire was composed of six questions. The results from the questionnaire are presented below.

Question 1*: Was Binary Blaster easy to use?*
100% of the students responded positively to this question and most of their comments were that the game is "very user friendly," "extremely easy to use and simple to play," and "very intuitive."

Question 2: *Did you enjoy playing Binary Blaster?*
100% of the students responded positively to this question. Some of the students' comments include:

- "I did enjoy the game but it was kind of easy. If I do ever play again I will probably try for more bits or something other that traditional binary."
- "I did enjoy playing the game, I got excited while in the middle of it ☺."
- "It reminded me of tetris for computer nerds."
- "The game was an enjoyable way to learn binary, twos complement and excess notation. I can use it as a study aid as well."

Question 3a: *Do you think Binary Blaster was effective for your learning/understanding of number conversion?*
100% of the students responded positively to this question commenting that the game was "very effective," as well as helpful for pattern memorization, for "getting a 'feel' of binary numbers," or "thinking 'on the fly' for binary conversion." One student also commented that "the game made learning the conversions fun, since it challenged you to convert the number in a short time period."

Question 3b: *If yes, in your opinion what aspect of the game was the most effective?*
Students' responses were related to either the "speeding process," the repetition and pattern memorization, or the fun aspects of the game. Some of the students' comments were:

- "The game 'forces' you to convert numbers in a short time period. If you can do it fast enough, you

get a point, if not you lose a point. It made it a competition with yourself to see how many points you could get before stopping."
- "It helped with memorization of patterns."
- "It kind of 'forces' the player to start remembering the base bit patterns and what they add up to."
- "It helped me recognize different patterns in the binary numbers in order to convert them to decimals."
- "It got me practice in a fun way."

Question 4: *Is there something that you did not like about Binary Blaster?*
75% of the students could not find something that they did not like about the game and 25% of the students complained about the speed variations in the game. These students' comments were:
- "I didn't like that after I had gotten so many answers right, if I missed one it wouldn't really slow down. It kind of created a negative cascading effect on my score. After I would miss one or two it would just be hard for me to get back into doing the binary."
- "I liked everything about the game, if I had to gripe about something it would probably be the number's drop speed. With larger bits, it became very difficult to compute the number before it reached the bottom."

Question 5: *Do you have any recommendations/comments to improve Binary Blaster?*
37.5% of the students had no recommendations and they commented that they like the game as is. The remaining 62.5% provided the following suggestions:
- "Add a menu to switch mid-game to two's complement. Also, it should slow down sooner, like when you miss one or two in a row."
- "I think the decimal number should be shown (maybe along with the binary match, when you get a correct answer). For reinforcement."
- "Add a pause feature."
- "Mainly just scale the drop speed with the number of bits that have been selected. Also maybe scale it with the type of notation chosen. This would allow more time to compute harder values, or less time for very simple values."

Question 6: *Would you recommend Binary Blaster to a friend? Why yes (why no)?*
100% of the students responded that they would recommend Binary Blaster to a friend who would like to practice binary conversions and they indicated that the game provides a good and fun way for students to practice binary conversions and "master bit patterns." In addition, one of the students provided the following comment: "I actually am thinking of using it over the summer so that I can teach some of my friends binary :) This is a fun interactive way to learn binary and I think that if you can associate fun with binary you're basically winning at teaching."

4 Conclusion and Discussion

In this paper, we have presented an electronic game (Binary Blaster) with which students can practice number conversion (decimal to binary). We also presented a small exploratory survey on students' perception of the game. The overall feedback from the students was positive with them indicating that they like the game, that the game is easy to use and fun to play, and that they feel the game was effective for their learning/understanding of number conversions. The results of this survey are encouraging for further investigation and continuation of the project.

One of our main future goals is to incorporate Binary Blaster into the instruction of the Introduction to Computer Science course and conduct a more comprehensive study with students to identify whether Binary Blaster is effective for improving students' knowledge and understanding about number conversions. Students' perspective of the game is also an important aspect for future developments and we are planning to conduct a study with undergraduate computer science and engineering students to collect additional feedback in regards to the functionality of the game as well as its appealing factors to the students. Based on the results from the exploratory survey presented here, we will update the "speed" feature of the game to slow down the "falling process" if students get one conversion incorrect, rather than waiting until they have five incorrect responses.

As part of our future work, we are also planning to include: (a) a collaborative mode to the game in which teams of two or more students may work together toward a common score, (b) an option to convert in the opposite direction (where a binary number appears in the falling shape and the numbers appearing at the bottom are displayed in decimal), and (c) an option to use other base systems such as hexadecimal and octal.

5 References

[1] A. D. Patterson, L. J. Hennessy. "Arithmetic for Computers," *Computer Organization and Design: The Hardware/Software Interface.* Morgan Kaufmann. Publishers. (2007).

[2] A. Ireland, D. Kaufman, & L. Sauvé. Simulation and Advanced Gaming Environments (SAGE) for Learning. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp.2028-2036). (2006). Reeves, T. & Yamashita, S., Eds. Cheasapeake, VA.

[3] Computing Curricula 2001. The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, Available at: http://www.computer.org/education/cc2001/final/index.html

[4] C. Petzold. “Alternatives to Ten. Code: The Hidden Language of Computer Hardware and Software,” *Microsoft Press*, (2000).

[5] D. Oblinger. “Simulations, games and learning,” *EDUCAUSE Learning Initiative*, pp. 1-6 (2006). Available at: http://net.educause.edu/ir/library/pdf/ELI3004.pdf

[6] G. J. Brookshear. “Data Storage. Computer Science: An Overview.” Pearson Education, Addison-Wesley. 2009.

[7] K. Becker. “A Multiple Intelligences Approach to Teaching Number Systems,” in *Proceedings of 2003 ITiCSE conference*, pp.250, (2003).

[8] K. Squire & H. Jenkins. “Harnessing the Power of Games in Education.” *InSight*, 3(5). Institute for the Advancement of Emerging Technologies in Education (IAETE). (2003). Retrieved July 10, 2010, from. http://website.education.wisc.edu/kdsquire/manuscripts/insight.pdf

[9] L. A. Steen. “Twenty Questions about Mathematical Reasoning,” in L. Stiff (ed), *Developing Mathematical Reasoning in Grades K-12*, pp. 270 – 285, (1999). Reston, VA: National Council of Teachers of Mathematics.

[10] M. Mayo. “Games for science and engineering education,” *Communications of the ACM*, 50 (7), (2007).

[11] M. Panoiu, A. Iordan, C. Panoiu, L. Ghiorghioni. “Educational Software for Teaching the Basics of Computer Science,” *WSEAS Transactions on Advances in Engineering Education*, 12 (6), pp. 238-243, (2009). Available at: http://www.wseas.us/e-library/transactions/education/2009/89-105.pdf

[12] M. Prensky. “Digital game-based Learning.” McGraw-Hill, NY. (2001)

[13] Pygame: http://www.pygame.org

[14] Python: http://www.python.org

[15] R. Turban. “Approaches to Implementing and Teaching Human Computer Interaction,” in the *Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC’03)*, pp. 81-84, (2003).

[16] R. Van Eck. “Digital game-based learning: It’s not just the digital natives who are restless.” *EDUCAUSE Review*, 41(2), 16-30. (2006).

[17] Tetris game: http://www.tetris.com/

[18] Wikipedia, Numeral Systems. Retrieved on May 15th from: http://en.wikipedia.org/wiki/Numeral_system

6 Acknowledgements

The authors would like to thankfully acknowledge the contributions of Kyle Schulz, David Danford and Heidi Lewis to the development of Binary Blaster.

Education in Clouds

Nedzad Mehic[1] **and Elyas Kashfi**[2]
[1] New York Institute of Technology-NYIT, Manama, Bahrain
[2] New York Institute of Technology-NYIT, Manama, Bahrain

Abstract - *Colleges and universities frequently maintain and upgrade their software and hardware to provide the best possible teaching and research environment. Usually these activities spend a significant portion of the IT budgets on these organizations. Keeping up with the growing costs has become more difficult due to the current global financial situation. The paper describes one possible solution to this problem by integrating the developed Educational Cloud Computing Model based on virtualization and cloud computing into an institution's curriculum. The Model provides an environment that is easier to maintain, more flexible to reconfigure, and with reduced operating costs. It has been built around the Microsoft® Hyper-V Servers 2008 R2, Linux Centos host OS, and the HyperVM virtualization management software. The pilot system has been successfully tested in the undergraduate and graduate courses such as programming languages, OS and OS security, networking, web technologies, and databases*

Keywords: cloud computing, hypervisor, virtualization, virtual machine

1 Introduction

In the past, most computation resources were served from dedicated machines running specific applications. As organizations grow or as computing resource demands have increased, problems may arise in terms of capital costs, resources capacity, deployment overheads amongst others. About 70% of a typical IT budget in a non-virtualized datacenter goes towards maintaining the existing infrastructure [24]. These concerns are more apparent in an academic institutions where issues such as costs (hardware and personnel), flexibility (quickly deploy or install software desired) and resource management are important in IT infrastructure. Academic institutions need to preserve balance between effective and efficient operations of computing infrastructure. However, it is very difficult to implement and maintain such environment. For example, it takes time to install and configure/reconfigure requested software on each computer lab. Additionally, some software may require elevated user privileges to run. Usually, it is not a good idea to give students such privileges. One way to solve these problems is to deploy an educational cloud computing platform based on virtualization technique. New technologies such as cloud computing, offer better methods in delivering computational resources on demand via a computer network as shown in Figure 1 [23].

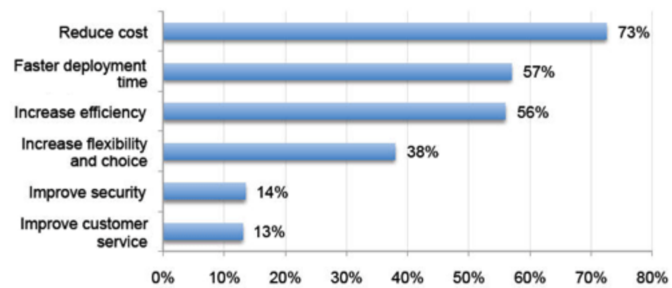


Figure 1 Migrating of corporate IT to cloud computing environments

2 Virtualization

Each virtual machine can interact independently with other devices, applications, data, and users as though it were a separate physical resource. Virtualization infrastructure shares physical resources of multiple machines across entire infrastructure. The Virtual Machine-VM, Virtual Memory Manager-VMM, and hypervisor or host OS are the minimum set of components needed in a virtual environment. Virtual machines can move and a single virtual machine that was once physically close to the data can theoretically be moved to a server anywhere on a global network. Because of the easy relocation, virtual machines can be used in disaster recovery scenarios [4].

The virtualization manager, commonly referred to as a "hypervisor," creates "virtual machines" that operate in isolation from one another on a common computer, Figure 2 [8]. The hypervisor allows different OSs to run in isolation from one another although they are computing power and storage capability on the same computer. This technique supports concepts like cloud computing.

Each virtual server acts just as a physical server multiplying the capacity of any single machine and extending server capabilities. In an advanced form, virtualization software called the hypervisor creates a layer between virtualization server and native hardware [6]. Hypervisor can run virtual instances of several different OSs at the same time and the virtual servers need not be aware that are running in virtualized environment.

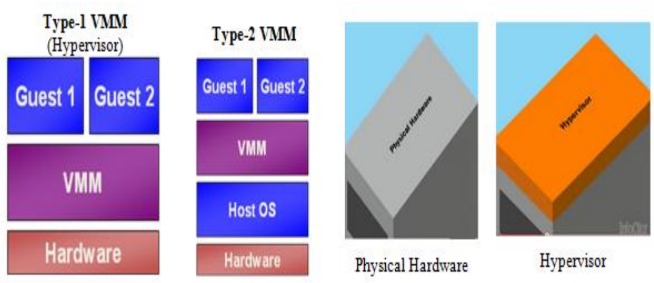


Figure 2 VMM, physical hardware, and hypervisor

3 Cloud Computing

Cloud computing has common elements with client-server model (service supplier and service user and peer-to-peer distributed model that does not require central supervision.The three main characteristics of cloud computing are [1].

1. The illusion of infinite computing resources available on demand
2. The elimination of an up-front commitment by cloud users, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs
3. The ability to pay for use of computing resources on a short-term basis as needed and release them as needed.

The main cloud computing features are on-demand access, elasticity, pay-per-use, connectivity, and, resource pooling [9]. Cloud computing models types are public cloud (cloud computing services used by the general public), private (usually designed and deployed on the premises of the organization that will make use of cloud resources) and hybrid (use both public and private clouds).

3.1 Cloud Computing and Academia

For academia, the cloud-powered system for higher education lets student and faculty access to file storage, databases, and other university applications anywhere, on-demand [17]. Cloud computing enables institutions to teach students in a new, different ways and help them to manage projects and course work. Figure 3 shows users of IT services at a typical university without and with support of cloud computing technology [19].

Cloud computing make available one or more layers that are provided "as a service" to end users. These are: Infrastructure as a Service-IaaS, Platform as a Service-PaaS, and Software as a Service-SaaS. Figure 4 shows a cloud computing model used in at a typical university [10]. This model offers access to application from anywhere, software free or pays per use, 24 hours access to infrastructure and content, increased openness of students to new technologies. The problems of this model are that not all applications run in a cloud, risks related to data protection, security and accounts management, lack of organizational support, software licensing, and intellectual property issues.

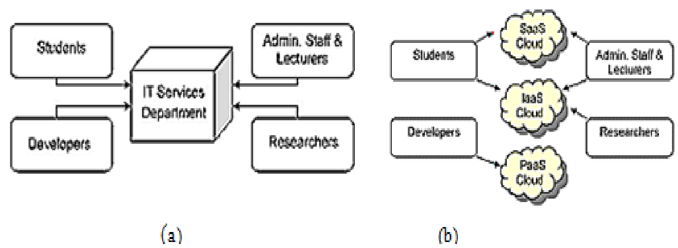


Figure 3 IT services users at a typical (a) and cloud computing (b) based university

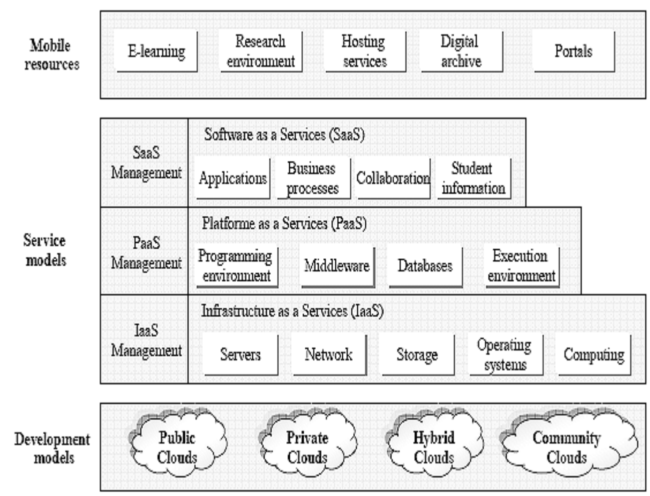


Figure 4 University based cloud computing architecture

Cloud computing represents an interesting model for developing nations, where the ability to access resources has often been limited and building a robust IT infrastructure can be difficult to achieve [3]. For example, IBM has established cloud computing centers in Beijing, China; Bangalore, India; Hanoi, Vietnam; Sao Paulo, Brazil; and Seoul, South Korea [5]. Cloud computing model is based on system modularity so organizations can break down services and systems into smaller components, which can function separately or across a widely distributed network. It is important to say that there are some opponents of the cloud computing model. Richard Stallman, creator of the GNU operating system and founder of the Free Software Foundation, calls cloud computing a trap aimed at forcing people to buy into locked, proprietary systems that are likely to prove costly in the future [12].

4 Virtualization and Cloud Computing Security

Many of the security risks are related to the cloud providers Figure 5 [23]. The major virtualization platform vendors, VMware, Xen (now Citrix), and Microsoft, have all had several vulnerabilities over the last few years [18]. This includes:

1. Host/Platform security
2. Hypervisor security
3. Securing communications
4. Security between host/guests

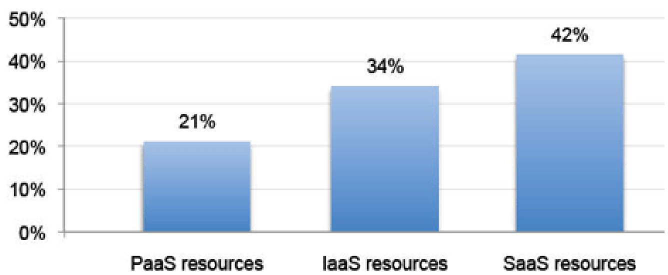


Figure 5 Cloud computing providers are most responsible for security

The virtual machines powering the cloud platform are susceptible to vulnerabilities. Figure 6 [9] shows basic VM system vulnerabilities.

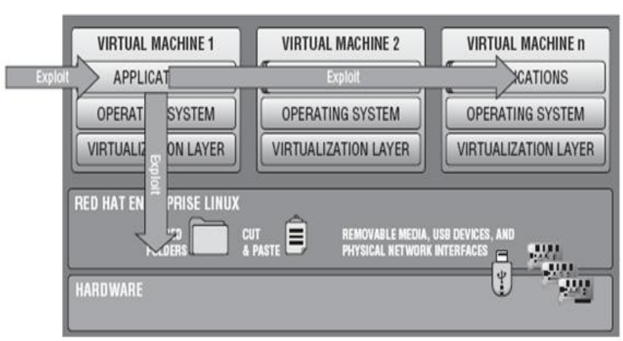


Figure 6 Basic VM system vulnerabilities

It is always a good practice to keep hypervisors (and kernels in the case of OpenVZ) constantly updated along with other system software. Users with access to computing resources must be given proper authentication and specific roles, for example giving SSH access to a user unfamiliar on how to use a terminal. Some threats to virtualized systems are general in nature, such as denial-of-service, or DoS, attacks others are specific to virtual machines. Many VM vulnerabilities stem from the fact that a vulnerability in one VM system could spread to the underlying hypervisor and, ultimately, to the systems of other customers. A one cloud computing customer could download a virus (e.g. that steals user data) and then spread that virus to the systems of all the other customers.

One of the main problems in adopting cloud computing technology in higher education is security. University databases are covering sensitive data on academic grades, health records and financial aid, among other things and are subject to different country laws. Certain countries have very strict rules about cross-border transfers of personal information and complying with those rules can be challenging in the virtual world of the clouds [17].

5 Project Development Concepts

The aim of this project is to implement an educational cloud with all features mentioned before. The end result is an environment that consists of virtual machines (with configurable resources) running a variety of operating systems (with or without preinstalled software), that were deployed and running from a centralized management control panel (HyperVM). The virtual machines are based on the popular software virtualization hypervisor known as OpenVZ.

The project has had two phases. In the first phase, the team developed a feasibility study on technical, organizational, and economical aspects of using cloud computing technology to develop an advanced computing environment. After the Study was accepted, a demo a system was developed with multiple Linux distributions accessible to clients over a network. Additionally, the platform should allow future growth by adding more nodes and provides an open API for applications development. Oracle's VirtualBox was a sandbox environment installed on a Windows 7 computer without hardware virtualization support. The software used was the cloud ready, Ubuntu 10.04LTS utilizing the Eucalyptus Cloud Computing Platform installation option. The setup required two different virtual machines running Eucalyptus, one that contained the global controllers (cloud, cluster and storage controllers), and another that operated as the node that runs instances. Although the installation and deployment was rather simple, the overall system suffered from instabilities and many configuration setbacks. Although the phase I has taken lot of time, it was a very useful exercise to learn more about new technologies.

The objective of the second phase was a development a pilot system of the Educational Cloud Computing model. We've decide to use powerful Hyper-V Windows 2008 Server R2 enabled servers as our working platform. The host operating system of the platform is CentOS, the hypervisor is OpenVZ, and the management interface is LxCenter HyperVM.

5.1 Project Development Details

Research was made in order to implement a pilot project that makes use of cloud computing technology in technology intensive schools such as those of engineering and computer science. Problems faced by the IT department included the limited number of staff, the large number of machines to manage and deploying specific software packages. By consolidating services, it was expected that a cloud computing platform would decrease associated costs of deploying, running and maintaining IT resources.

5.2 Microsoft Hyper-V Platform

Microsoft's hypervisor, Hyper-V, forms the core of Educational Cloud and is available on Windows 2008 Server R2. Configuration and setup of Hyper-V and virtual machines are done either through a command line interface or, by making use of Microsoft Management Consoles-MMCs setup on machines that connect remotely to the server. Unlike OpenVZ and Xen, Hyper-V supports installation of guest OS from an ISO file or directly from physical medium such as a CD or DVD. Microsoft Hyper-V also allows administrators to add and remove virtual components.

One of the unique aspects of Hyper-V is dynamic memory allocation support. Dynamic memory allocation reduces the need of administrators to estimate usage of memory while the virtual machine is running. As more memory is required, Hyper-V automatically allocates without the need of restarting the guest OS to recognize the additional memory. Currently only Windows Server and Windows 7 guest OS are able to use this feature [11]. To make use of dynamic memory allocation several parameters must be configured: startup RAM (minimum allocation), maximum RAM (maximum allocation), memory buffer (memory percentage allocated as a safety margin) and memory weight (prioritizing allocations over several VMs) (Microsoft, 2010).

Hyper-V is also able to achieve high availability of virtual machines by making use of its failover clustering feature which requires a minimum of two physical nodes and a high speed storage device available on the network (using iSCSI for example). In clustering, virtual machines are kept in sync, in the event of a failure of one virtual machine or physical node; the workload will be transparently distributed to the other(s). In the Model, a Hyper-V virtual machine will power the middleware software, the CentOS operating system.

5.3 CenOS

In our solution, the CentOS OS was installed and configured to be apart of the Hyper-V virtual machine to run OpenVZ containers and make use of LxCenter HyperVM. CentOS is an ideal choice to power the cloud because it is simple to install and use but still maintains compatibility with the major Linux distribution. It is important to note that the updates released by Red Hat usually become available soon on CentOS since it "aims to be 100% binary compatible" [2] with little to no difference.

CentOS is community driven, administrators may find the information and support they require from mailing lists and forums rather than a technical support representative (as in Red Hat). While educational institutions may prefer to have enterprise support for their operations, the CentOS community provides a great wealth of information regardless of the issues. Due to its close compatibility and reliability to Red Hat, CentOS is the number one server OS used by webservers around the world taking a 29.2% share of Linux [21].

OpenVZ has been was chosen to be our virtualization environment because it is an open source container-based virtualization solution built on Linux [13]. It creates isolated, secure containers where containers can be rebooted independently and have root access, users, IP addresses, memory, processes, and, and files.

The main disadvantage of using OpenVZ is that the same kernel powering the operating system (CentOS in the platform) is also used to power containers. As a result only Linux can be virtualized. The default CPU scheduler of OpenVZ assigns to each container a fair share of CPU time. Resources allocation such as memory, network limiting, simultaneous processes etc… are controlled through "user bean counters" [7]. A container user may review the allocation by reading the "/proc/user_beancounters" file from within the container. Traditionally administrators set these limitations through commands, however, a virtual machine manager such as HyperVM provides a point-and-click solution. Rather than utilizing "images", as known in other virtualization technologies, OpenVZ makes use of "templates" for launching virtual machines. Simply put, a template is a tar gzipped copy of the Linux file system for each respective distribution. Templates are simply downloaded from the site http://wiki.openvz.org/Download/template/precreated to the "/vz/template/cache" directory and are ready for use. The figure below shows some of the templates under HyperVM.

Migrations and backups are simplified in the event of a disaster and because it involves copying the contents of the containers and adjusting usually a single configuration file. Compared to Xen (another virtual environment offered by Citrix company), OpenVZ containers demonstrate low overhead and high performance in file system operations [22]. A HP (Hewlett Packard) research has also shown that "CPU consumption for Xen is roughly twice that of the base system or OpenVZ" [15]. OpenVZ provides two types of NICs, "veth" (virtual Ethernet) and "venet" (virtual network). The main difference between them is that "veth" interfaces are directly bridged to the physical NIC making it less secure, but necessary in applications that require a valid MAC address [14]. The Educational Cloud Model uses "venet" interfaces to simplify network management and enhance security.

Size	File Name ▲
164.2M	centos-5-i386-afull.tar.gz
434M	centos-5-i386-gnome.tar.gz
373.9M	centos-5-i386-hostinabox576.tar.gz
132.2M	debian-5.0-x86.tar.gz
155.9M	debian-6.0-eyeos_1.9.0.2-SVN-7072_i386.tar.gz
74.3M	debian-6.0-i386-minimal.tar.gz
192.5M	fedora-13-x86.tar.gz

Figure 7 List of templates under HyperVM

It is important to understand that once a template is installed, the corresponding Linux does not have graphical interface and commands are entered through the shell. In order to install GUIs, it is necessary to install two types of software: the desktop environment and Virtual Network Computing-VNC) server. Each operating system has a similar installation process. For example, the method used for Red Hat based OS such as CentOS or Fedora is as follows:

yum groupinstall "GNOME Desktop Environment"
Install all components required to provide a GNOME environment.
yum install vnc-server
Install the VNC server.

5.4 OpenVZ Management using LxCenter HyperVM

The final piece of the Education Cloud Model the LxCenter HyperVM provides administrators with a simple and graphical method of managing instances and associated resources, most importantly through a browser for global accessibility. Although there are a wide variety of paid and free control panels, LxCenter's HyperVM provides a free to use panel with community driven support. HyperVM is open source, allowing developers to create applications and provide a better security model. Additionally, open source software is perfect for educational institutions that would like to deploy clouds in the most cost effectively and flexibly. HyperVM supports RHEL5 (Red Hat Enterprise Linux), CentOS 5 and Fedora 2 to 6. The installer provides the option of installing either OpenVZ or Xen.

The HyperVM manager comes with many additional features that provide more control and ease of use of virtual environments. These features include a unified management portal that is able to control multiple installations of HyperVM and a web based GUI to manage all virtual machines and core HyperVM functions. Pools of IP addresses can be created that can be attached to a virtual machine.

Plans, client management and DNS are also supported within HyperVM as illustrated in Figure 8. Transfer allowances or bandwidth can be limited on a per virtual machine basis. Resources management is simplified in HyperVM, instead of using commands to set resources limits to a virtual machine, administrators can control limits through the web interface. A complete API is included in HyperVM which can accessed either through an HTTP request or directly on the command line prompt of the running host node running HyperVM.

Important part of the phase II was security risk analysis. Each of the components from Microsoft Hyper-V to a service running within a cloud is vulnerable to attacks. The node running Hyper-V has been set to check and install updates from Microsoft every day to keep the environment bug free and secure. The CentOS installation within the virtual machine, goes through a "yum update" whenever security errata are published by Red Hat and subsequently introduced by the CentOS community. LxCenter's HyperVM has had its share of security vulnerabilities in the past, but after its release as open source important patches were introduced to bring it up to par for use within a production environment.

Finally, there are security concerns with the OpenVZ, templates because their updates are not done automatically and should be done using a distribution's package manager to performed this task after launch. Additionally, only official OpenVZ released images should be used to protect the system.

Figure 8 Some of the VM resources configurable in HyperVM.

5.5 Browser and Security

One of the main features of the Educational Cloud Model is the ability to connect to a virtual desktop using a Java enabled web browser. The HyperVM interface is also accessible through a web browser by administrators. It is therefore paramount to keep web browsers updated constantly. Users today have a choice of web browsers ranging from Google Chrome, Mozilla Firefox, Opera and many more. Cloud computing services may become vulnerable or possibly not work simply because of a client web browser. A web browser's developer may have ceased introducing patches to a branch of the browser, launched a new updated branch with many fixes, but user adoption is low thus they continue to use a vulnerable browser.

Today's browsers also allow multi-tasking by implementing multi-tabbed interfaces and in the past vulnerabilities took advantage of such features. Other vulnerabilities come from additions used to enrich the Internet experience such as JavaScript and Flash. Malicious code can be injected into pages by taking advantages of flaws in such plug-in (known as cross-site scripting attacks, XSS). It is therefore important to educate the user as well to keep their browsers up to date and use good security practices when accessing resources through a web browser.

6 Cloud Computing and Legal Issues

In situations where a cloud computing platform is built in house (in the case of private clouds), legal issues surrounding data stored comes under control of the organization that built the infrastructure. As a result organizational policies regarding data management is enforced in the platform. Additionally, data will also need to be stored in accordance to local laws and regulations, which is trivial in private clouds since it is known beforehand where and how data is going to be stored. Legal issues do arise when certain software cannot be licensed to virtual machines, especially legacy software that is not actively developed.

In public clouds (and data stored in hybrid clouds) however, legality becomes much more complex. Firstly, the cloud provider may not disclose where the data is going to be stored. This may not be important for some data but for sensitive data it is paramount to know where the data will reside. Along with location come local laws and regulations, such is the case when it comes to deciding courts in lawsuits between the customer and a cloud provider. Providers must provide some form of guarantee on the service they are providing to the customer. This is usually done through a Service Level Agreement- SLA that addresses: application security, intellectual property protection, termination, compliance requirements, customer responsibilities, performance tracking, problem resolution amongst others [9]. Organizations must be very careful in their intentions of moving some (or all) of their infrastructure to the cloud to avoid legal implications as a result.

7 Conclusion

The paper describes implementation and deployment of a new type of a computing system for higher education based on virtualization and cloud computing with increased overall IT effectiveness at reduced cost. The pilot system allows students and faculty to use different Linux clouds and their applications on-demand. It has been successfully tested in both undergraduate and graduate programs. The users can use major Linux distributions and their applications with reduced downtime due to improved maintenance and keeping OS and software packages constantly updated and patched.

Future work will also include expanding the system to other Schools within the University and also implementation of the Xen hypervisor for running other OS's templates.

8 References

[1] Armbrust, M. et al. (2009). *Above the Clouds: A Berkeley View of Cloud Computing*, Technical Report No. UCB/EECS-2009-28, University of California at Berkeley. Retrieved from: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf.

[2] CentOS. (2009). *Purpose of Centos*. Retrieved from: http://www.centos.org/modules/tinycontent/index.php?id=3

[3] Greengard, S. (2010). Cloud Computing and Developing Nations. *Communications of the ACM*, 53(5), 18-20.

[4] Hansen, G. (2010). *How Virtualization and Cloud Computing Will Change Security*. CTOEdge. Retrieved from: http://www.ctoedge.com/content/how-virtualization-and-cloud-computing-will-change-security.

[5] IBM. (2010). *IBM Launches Academic Cloud to Speed Delivery of Technology Skills to College Student*. Retrieved from: http://www-03.ibm.com/press/us/en/pressrelease/29367.wss.

[6] InfoWorld. (2007). *InfoClipz: Server virtualization*. Retrieved from http://www.infoworld.com/ifwclassic/video/archives/2007/02/server_virt.html.

[7] Kolyshikin, K. (2006). *Virtualization in Linux*. Retrieved from: http://ftp.openvz.org/doc/openvz-intro.pdf.

[8] Kieffer, M. (2010). *Windows Virtualization Architecture*, Microsoft. Retrieved from: http://download.microsoft.com/download/5/b/9/5b97017b-e28a-4bae-ba48-174cf47d23cd/vir040_wh06.ppt.

[9] Krutz, R. & Vines, R. (2010). *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Indianapolis, IN, USA: Wiley Publishing Inc.

[10] Mircea, M. & Andreescu, A. (2010). Using Cloud Computing in Higher Education: A Strategy to Improve Agility in the Current Financial Crisis, *Communications of the IBIMA*, 53(5), Retrieved from: http://www.ibimapublishing.com/journals/CIBIMA/2011/875547/875547.pdf .

[11] Microsoft. (2010). *Hyper-v dynamic memory configuration guide*. Retrieved from: http://technet.microsoft.com/en-us/library/ff817651(WS.10).aspx.

[12] Stallman, R. (2008). *Cloud Computing is Stupidity*. NewsCnet, Retrieved from: http://news.cnet.com/8301-1001_3-10054253-92.html.

[13] OpenVZ. (2011). *What is OpenVZ?*, Retrieved from: http://old.openvz.org/.

[14] OpenVZ. (2010). *Differences between venet and veth*. Retrieved from: http://wiki.openvz.org/Differences_between_venet_and_veth

[15] Padala, P. (2008). *Performance evaluation of virtualization technologies for server consolidation*. Retrieved from: http://www.hpl.hp.com/techreports/2007/HPL-2007-59R1.html.

[16] Prism Microsystems. (2010). *2010 State of Virtualization Security Survey*. Retrieved from: http://www.prismmicrosys.com/documents/VirtualizationSecuritySurvey2010.pdf.

[17] Sasikala, S., Prema, S. (2010). Massive Centralized Cloud Computing (MCCC) Exploration in Higher Education. *Advances in Computational Sciences and Technology*, 3(2) 111–118. Retrieved from: http://www.ripublication.com/acstv3/acstv3n2_2.pdf.

[18] Shackleford, D. (2010). *An Introduction to virtualization security*. SANS. Retrieved from: http://www.net-security.org/article.php?id=1397&p=2.

[19] Sultan, S. (2010). Cloud Computing for Education: A New Dawn?. *International Journal of Information Management*, 30, 109–116. Retrieved from: http://www.just.edu.jo/~amerb/teaching/1-10-11/cs728/8.pdf.

[20] Vogels, W. A. (2008). Head in the Clouds-The Power of Infrastructure as a Service. In *First workshop on Cloud Computing and in Applications* (CCA ’08).

[21] W3Techs. (2011). *Usage statistics and market share of Linux for websites*. Retrieved from: http://w3techs.com/technologies/details/os-linux/all/all.

[22] Walters, J., Chaudhary, V. (2009). A fault-tolerant strategy for virtualized HPC clusters. *Journal of Supercomputing*. 50(3), 209-239.

[23] Vizard,M. (2010). *The State of Cloud Computing Security*. CTOEdge. Retrieved from: http://www.ctoedge.com/content/state-cloud-computing-security?slide=3.

[24] VMWare. (2011). *Virtualization Basics*. Retrieved from: http://www.vmware.com/virtualization/virtual-infrastructure.html.

GIS Laboratory: New Trends in Research and Education in Environmental Science Based on Advanced Computing Tools

L. Matejicek[1]
[1]Institute for Environmental Studies, Charles University in Prague, Prague, Czech Republic

Abstract - *This contribution provides an overview of some of the recent research in environmental informatics involving spatio-temporal data processing. Implementation of cloud computing to teach Geographic Information System (GIS) and spatial sciences is described in the framework of a project focused on innovation of the GIS Laboratory. The new age of cloud computing is demonstrated on research issues dealing with environmental science including topics such as monitoring of air pollution with LIDAR, modeling of surface water pollution extended by data from remote sensing, optimization of municipal solid waste collection in urban areas based on linear programming, and modeling of vegetation development in spruce forests with a large set of ordinary differential equations. Other related domains such as image processing in remote sensing, and mobile GIS extended by global positioning system (GPS) are discussed in addition to spatio-temporal processing in the GIS.*

Keywords: GIS; Environmental Informatics; Environmental Modeling; Cloud Computing; Computer Learning Tools

1 Introduction

GIS (Geographic Information System) has been couched as one of the outstanding technologies of the last century. Now it is evident that recent developments in the geotechnologies of GIS and related disciplines such as remote sensing and global positioning system (GPS) have had a considerable impact on environmental research providing spatio-temporal data to enable the further understanding of environmental systems [1]. In the last decade the developments related to the fields of GIS, such as high-performance communication networks, data standards and interoperability, sensing technology and spatio-temporal modeling have resulted in revelation of new phenomena and posing of what might be termed next generation of environmental questions. It requires more support for the open exchange of data and software, which needs infrastructure for location based services.

The new age of cloud computing is rapidly emerging as a technology trend that almost every industry focused on GIS can leverage. The technology and architecture that cloud service offer for research and education are a key area of research and development for implementation of hardware, software and spatio-temporal data models in our GIS Laboratory. Among several variations on the definition of cloud computing, our laboratory is dealing with GIS research tools that are delivered on demand as a service via the Internet [2]. The evolution of cloud computing over the past few years indicates outstanding progress. However, if cloud computing is to achieve its potential, there needs to be a clear understanding of the various issues involved, both from the perspectives of the providers and the consumers of the technology. While a lot of research is taking place in the technology itself, there is an equally urgent need for understanding the business-related issues such as case oriented research projects and emerging virtual learning environments surrounding cloud computing. In this contribution, new trends in research and education focused on environmental science are introduced in the framework of educational establishments that continue to seek opportunities to rationalize the way they manage their resources [3].

1.1 Geospatial infrastructure

Think of our living environment as a spatial information system extended by temporal data gives us a new insight into phenomena and processes in the real world. Representations of real-world entities are drawn with point, line and polygon symbols and are also shown with imagery [4]. Points, lines, polygons, and images are organized into thematic layers in order to represent cities, river networks, lakes or aerial photographs and satellite images of the landscape. Thematic layers are the starting point for environmental modeling our world. They not only contain collections of similar entities, but also model their attributes, relationships, and behavior. The same entities used to create map layers are employed by ArcGIS to construct modern information systems. In ArcGIS, the geodatabase is the native data structure for storage and analysis of spatio-temporal data. A collection of thematic layers is represented by a set of thematic datasets. Datasets representing satellite and aerial photo imagery can be also stored in a geodatabase in order to be used as a foundation for other thematic layers or to represent continuous surfaces and thematic data. In the simple terms, the geodatabase has a data model that is implemented as a series of data tables holding feature classes, raster datasets, and attributes. Advanced data objects can add GIS behavior, rules for managing spatial integrity and spatial relationships of core features, rasters, and attributes. The geodatabase is managed by ArcGIS in order to provide editing workflows such as adding, deleting, and updating features and attributes [5]. There are three types of geodatabase in relation to the data storage within a single Microsoft Access, as a collection of binary files, and in the environment of relational databases systems.

1.2 Geospatial analysis and modeling

GISs offer a wide range of methods for exploration of spatio-temporal data. Exploratory spatial data analysis allows users to examine their data in many different ways from the global and local view [6, 7, 8]. In the past, it has been necessary to couple GIS with case oriented software tools designed for high performance in dynamic modeling [9]. But with the increasing power of GIS hardware and software, it is now possible to reconsider this relationship in the framework of conventional tools and advanced computing structures based on cloud computing and other related technology. Models based on spatio-temporal data are dependent on spatial and temporal resolution [10]. Both spatial and temporal resolution need to be appropriate to the real nature of the process being modeled [11, 12, 13].

New trends in research and education presented in this contribution are focused on environmental modeling supported by GIS, and partially by remote sensing and GPS [14]. A few research projects are introduced in order to demonstrate the implementation of cloud computing in the framework of environmental research and teaching of spatial sciences. They deal with monitoring of air pollution with LIDAR, modeling of surface water pollution extended by data from remote sensing, optimization of municipal solid waste collection in urban areas based on linear programming, and modeling of vegetation development in spruce forests with a large set of ordinary differential equations.

2 The GIS Laboratory

The GIS Laboratory was established at Charles University in Prague in the framework of the research project supported by the Ministry of Education, Youth and Sports. The research project was maintained for 4 years (1997-2000) and afterwards extended by other grants and research projects. Our research themes, spanning the whole of environmental science from the spatio-temporal modeling point of view, have been running for a few years. Our research is closely integrated with our teaching, providing a wide range of courses. The staff of the laboratory participates in a full range of degree programs from the Ph.D. to the M.Sc. and the B.Sc. in environmental science and natural resource conservation. The interdisciplinary programs established to increase mutual understanding among the people of Europe include Erasmus Student Network (ESN) and other prestigious awards programs. Our students are drawn from the Czech Republic, and partly from the European Union and other countries.

2.1 Innovation of the GIS Laboratory

The increased awareness of GIS technology within and outside our GIS Laboratory has increased demand for the laboratory's teaching and research services, so much so that the GIS Laboratory was greatly renovated in the framework of the project supported by the Ministry of Education, Youth and Sports (FRVS 131/2009A/b). The project was maintained for one year (2009). The current objectives of the GIS Laboratory include quality teaching within the Ph.D. to the M.Sc. and the B.Sc. programs. The lectures and exercises offered by our laboratory cover a wide range of subjects such as Environmental Informatics; Environmental Modeling; Advanced Methods in Geoinformatics; Applied GIS and Remote Sensing; Energy in Nature and Society; Applied Ecology; Advanced Methods of Environmental Research; and other related subjects, Fig. 1. Our teaching providing a wide range of other stimulating courses is closely integrated with our research that involves our academic and research staff and postgraduate students in numerous inter-organization collaborative projects. The members also organize seminars, workshops and discussion groups that can be supported by new advanced computer systems based on new technology and architecture of cloud services.

Fig. 1. Advanced Methods of Environmental Research: terrain measurements with GPS and mobile GIS.

New trends in research and education in our GIS Laboratory are actually supported by a two-socket, Intel-based server (2x Intel Xeon E5530 processor, 32GB memory, 2x 450 GB RAID 1, 6x 1 TB RAID 5, Windows Server 2008, x64 edition, Hyper-V), which is suitable for medium businesses. New virtualization technology enables us to deliver more advanced capabilities to our spatial processing in environmental science. It also offers to take advantage of the cost savings of virtualization in order to consolidate multiple server roles as separate virtual machines running on a single physical machine, and leverage the power of x64 computing. In addition to primary server roles (domain management, file and print servers, FTP and WWW servers), the GIS server such as ArcGIS for server is installed in this virtual environment, which gives us the ability to manage and distribute GIS services over the Internet to support desktop, mobile and Web applications. The virtual GIS server is complemented by desktop ArcGISs and relational database systems installed on standalone virtual computers. Thus, our users are able to share environmental data more easily, and scientists have quick access to data for viewing and analysis in GIS. These services are transitioning to cloud computing, wherein technological capabilities are commonly maintained off premises and delivered on demand as services via the Internet. The cloud GIS costs less, is always available, has faster application delivery, is flexible, and has improved business continuity. ArcGIS server services and desktop applications are extended by software tools focused on image classification and data processing of terrain environmental measurements and observations. It enables to standardize the spatio-temporal data processing and sharing geospatial data among users. A basic schema of the computer tools in the innovated GIS Laboratory extended by the cloud computing is in Fig. 2.

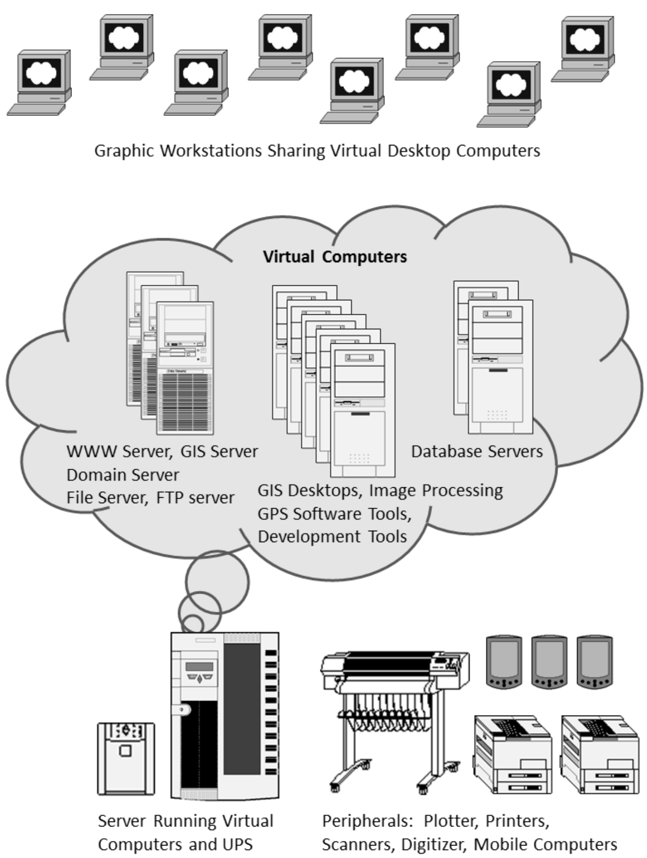


Fig. 2. GIS Laboratory relationship with the cloud computing deployment models: a basic schema of computer tools for education and research.

2.2 Environmental research projects

In order to explore environmental pollution and stress in a more complex way, data processing and environmental modeling include a wide range of techniques that overtax many of present data structures and algorithms [15, 16, 17, 18]. In the past, air pollution, surface water pollution, soil contamination, noise assessment, waste management and landscape protection were monitored in the local scales. Independent techniques were used for data processing and model predictions [19, 20, 21, 22, 23]. Nowadays environmental complex analysis of urban sites and rural areas dictates integration of the standalone data analyses and environmental models into complex systems. A number of case oriented solutions are described in scientific papers and research reports [24, 25, 26, 27]. The examples of spatio-temporal data processing and environmental modeling presented in the framework of lessons and exercises are based on selected research projects that are focused on monitoring of air pollution with LIDAR, modeling of surface water pollution, optimization of municipal solid waste collection in urban regions, and spatio-temporal modeling of vegetation changes in mountain forests.

2.2.1 Monitoring of air pollution with LIDAR

Extensive exploitation of natural resources followed by the increase in transportation infrastructure has negative impacts on the environment of a large part of the population primarily in the urban areas [28]. One of the main factors appears to be the quality of the air affected by anthropogenic pollutants influencing the formation of smog and acid rain. Exploration of these processes in the context of the geographic location and a spatial model description of the territory then becomes a necessity for obtaining more exact conceptions of the dispersion of pollution and chemical processes occurring in the atmospheric boundary layer. It is usually evaluated on the basis of direct measurements, which are limited due to economic reasons to individual observations in time and space. Remote sensing techniques get more information about dispersion of monitored air pollutants. Laser scanning techniques (Differential Absorption LIDARs) can be used for detection of atmospheric pollutants such as the nitrogen dioxide [29]. But new data processing tools are needed to provide exploratory spatial data analysis and model predictions. The measured concentrations of nitrogen dioxide by LIDAR in horizontal and vertical scans are registered in the beam files. Post-processing procedures are required to convert data to the GIS project that can manage spatio-temporal analysis, model predictions and visualization.

The ArcGIS was used to estimate the distribution of the pollutants in the vertical plane over the surface of the arterial road. The spatial interpolation by ordinary kriging was created in GIS [30, 31] and transferred to a vertical plane over the digital terrain model extended by urban buildings and vegetation. The GIS project contains a digital terrain model extended by urban buildings and vegetation, and other map layers focused on risk assessment of the area of interest. Spatial data and LIDAR measurements are managed with geodatabase, which provide the common data storage and management framework for ArcGIS. It can be leveraged in other GIS environments accessible through the cloud computers for exploration and processing. As an example, a spatial interpolation of nitrogen dioxide concentrations by ordinary kriging in a vertical plane over the digital terrain model, urban buildings layer and vegetation layer is in Fig. 3. Spatial interpolation created the continuous layer from 50 point samples, predicting the values of nitrogen dioxide concentrations (in the range from 5 μg/m^3 to 120 μg/m^3) over the surface of the arterial road in the City of Prague.

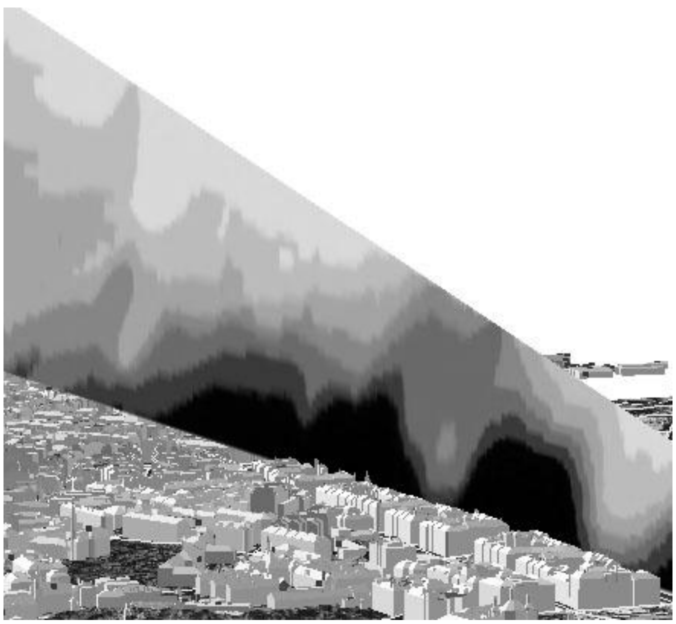


Fig. 3. The spatial interpolation of the nitrogen dioxide concentrations by ordinary kriging in a vertical plane in the range from 5 μg/m^3 to 120 μg/m^3 over the digital terrain model extended by urban buildings and vegetation (higher nitrogen dioxide concentrations are linked to darker shades).

2.2.2 Modeling of surface water pollution

Many surface waters are impaired by excessive nitrogen and phosphorus from point and nonpoint sources. Thus, the management to reduce watershed nitrogen, phosphorus and other pollutants is a priority, especially in regions of intensive human activities [32, 33]. Environmental models are useful tools for analysis of surface water quality, and for assessment of environmental management scenarios. An integrated approach for surface water quality modeling involves connecting geospatial data describing the physical environment with water quality analysis and water pollution process models. Time series of water quality data at monitoring points are needed in order to verify models. Advanced GIS extensions can provide a robust environment for integrating geospatial and time series data for water resources with environmental models for water flow and quality [34]. Nowadays, the advances in the computing tools and GIS-based systems greatly enhance the capacity for research scientists to develop environmental models due to the improved data management and rapid simulation tools that can be built into a GIS driver [35, 36]. Software tools for assessment of pollution sources are principally based on dynamic models that mostly include a few subsystems focused on soil erosion, sediment transport, and runoff phenomena such as Soil Water Assessment Tool (SWAT) [37].

The attached case study was focused on implementation of the compartment model that includes mass transport and accumulation of pollutants in each part of the catchment [38]. The satellite and aerial images complemented with GPS terrain measurements enabled to estimate point and non-point sources of pollution. The spatio-temporal model is able to predict the long-term impacts. It can be used to simulate at the catchment scale nutrients cycle in landscapes whose dominant land use is agriculture. A preliminary land use considerations is based on Normalized Difference Vegetation Index (NDVI) complemented by terrain observations. A numerical model is accessible through the cloud computers on the desktop GIS and the mobile GIS.

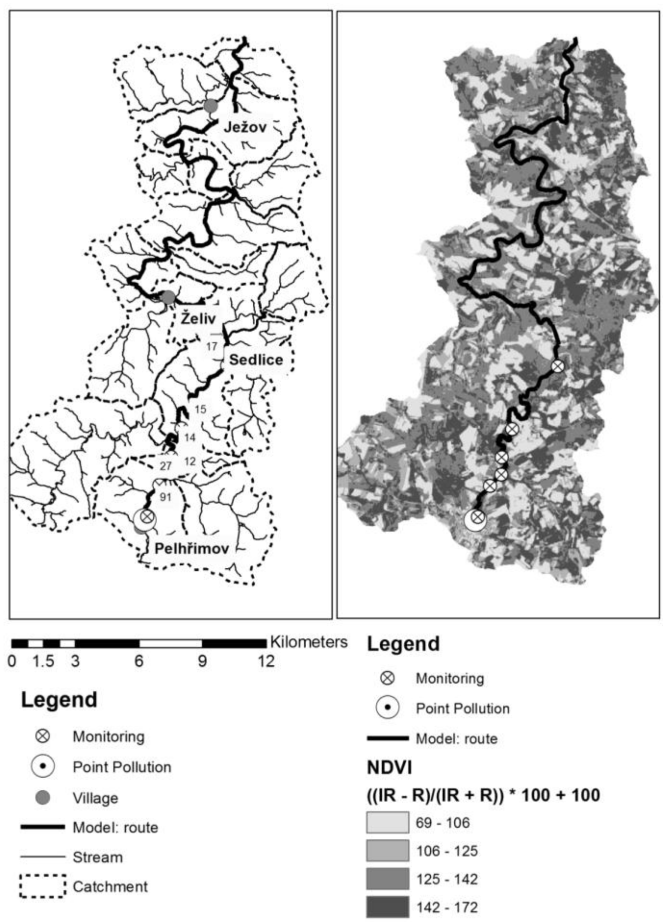


Fig. 4. The catchment illustrated by the thematic map layers and the NDVI image complemented by stream compartments.

2.2.3 Optimization of municipal solid waste collection in urban regions with linear programming methods

Solid waste management systems in Europe cover many of technological alternatives, economic instruments, and regulatory frameworks [39]. Various environmental, economic, social, and regulatory impacts in waste management practices not only complicate regional policy analysis, but also reshape the paradigm of global sustainable development. Systems analysis supported by GIS can harmonize these integrated solid waste management strategies, because system assessment tools and prediction models sharing spatio-temporal data and methods are able to handle particular types of problems. In order to overlap boundaries in waste management practices in European countries and unify systems analysis, more complex and widely used analytical tools based on GIS and cloud computing can provide opportunities to develop better tools for solid waste management strategies leading to conformity with current standards and future perspectives for both the industry and government agencies.

An attached case study is focused particularly on optimization of municipal solid waste collection in the City of Prague [40]. The ArcGIS was used to create a series of polygons representing the areas that could be reached from target points within specified road distances. The case oriented extension based on linear programming optimized transportation amounts between the resources and targets, which is displayed with origin-destination lines in Fig. 5. The spatio-temporal data are managed with geodatabase. The optimization tasks can be processed by case oriented ArcGIS modules that are accessible through the cloud computers. With the outlined study and other developments, it is believed that assessment tools should lead to improved insights and generate a suite of better management strategies.

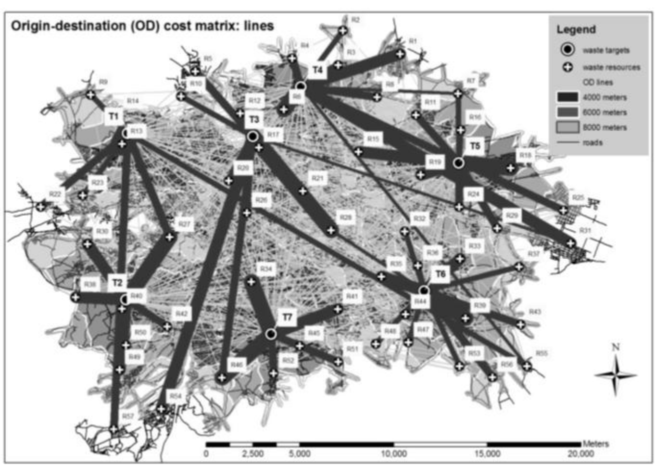


Fig. 5. The schema of municipal solid waste resources and targets extended by service areas for selected road distances (4000, 6000, 8000 meters), and by origin-destination lines (the line width symbolizes the quantity of the transported municipal waste from municipal solid waste resources to targets of waste disposal labeled by T1, T2, …,T7).

2.2.4 Spatio-temporal modeling of vegetation changes in mountain forest

Stress from air pollution has been caused degradation of forest ecosystems in Western and Central Europe for decades. In order to study subsequent regeneration, the relevant processes need to be quantified by spatio-temporal modeling in various spatial and temporal scales. Setting of the spatially dependent growth conditions is derived from a number of terrain observations that are converted into map layers such as the crown projection layer, the slope layer, the dead wood layer, the soil map, the ground vegetation layers of dominant species, and a wide range of other environmental properties. In spite of that many models studies have been performed on tree growth and regeneration for a few decades [41], the results are prone to limitations caused by the local conditions and the legacies of the past. Thus, the simulation results cannot be simply applied and a new spatio-temporal model is required to adopt local conditions. In order to provide exploratory spatial data analysis and model predictions, the GIS environment advertises its complex analytical tools focused on spatial data management and analysis, and development tools that can customize functions and create new case oriented applications for dynamic modeling.

A spatio-temporal model was developed in order to predict ground vegetation changes and tree regeneration in the mountain climax Norway spruce forest during the recovery period (1995-2006) [42]. The model was based on a set of ordinary differential equations that describe dynamic processes in microsites arranged in a grid (50 x 50 microsites) for each ground vegetation species and each age group of Norway spruce seedlings. A few case studies were developed to simulate various biotic and abiotic interactions. The area of interest was visualized with GIS in order to explore outputs of the dynamic model. The spatio-temporal modeling was implemented in the ArcGIS environment. A set of ordinary differential equations was solved in MATLAB extended by the parallel computing toolbox. A few virtual computers with described models managed by standard PCs are in Fig. 6.

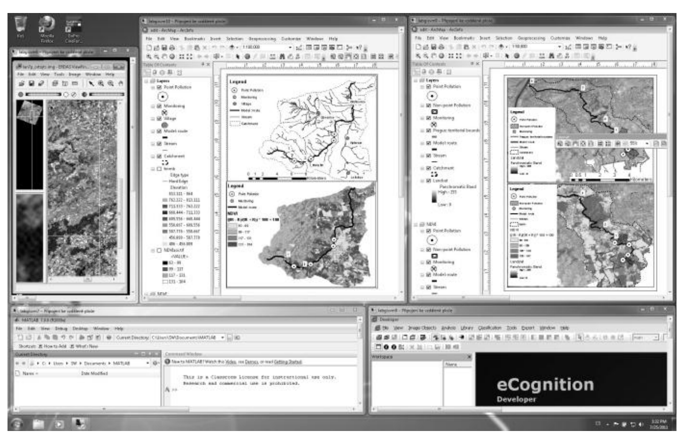


Fig. 6. Case studies and computing tools accessible through virtual computers managed by the standard PC environment.

3 Conclusions

Innovation of the GIS Laboratory significantly promoted new trends in education of environmental science. In addition to modernization of computer systems in the computer laboratory and the attached classrooms, new trends in environmental informatics and modeling were brought by advanced computing tools based on cloud technology and GIS. Our teaching providing a wide range of lectures and exercise was complemented by research tasks based on advanced software tools focused on data processing in GIS, remote sensing, GPS, and dynamic modeling. Our stimulating courses were more closely integrated with our research in environmental science. It promoted integration of our academic and research staff and postgraduate students in numerous inter-organization collaborative projects.

The cloud computing focused on spatio-temporal modeling in environmental science represents an evolution from existing technologies [43, 44, 45], catalyzed by high speed networks and new generations of computer tools that take traditional GIS operations into the world of interconnected data and more accessible software tools, which rationalize the way they manage their resources. In addition to GIS software and spatio-temporal data, case oriented software tools for image processing and classification can be accessible to a wide range of users sharing research in our GIS Laboratory. Data processing from GPS can be provided in a more efficient way in relation to GIS projects and image processing tasks.

Cloud computing in our GIS Laboratory represents an emerging computing paradigm which promises to provide opportunities for delivering a variety of computing services in a way that has not been experienced before. Thus, new research projects and education activities must be adapted to a more powerful and efficient computing environment. Like many new technologies and approaches, cloud computing is not without problems. There can be many concerns relating to its security and portability. The specific roadmap for environmental informatics dealing with GIS and cloud computing might be still unclear, but the fundamental economic and business forces that shape the computing industry point to a logical conclusion: GIS tasks and environmental models are general-purpose in nature, and therefore offer tremendous economies of scale if their supply can be consolidated for academic research. New trends in research and education are briefly demonstrated on selected research projects that can support research activities in a more efficient way through the cloud computers.

New trends in education based on described advanced computing tools will require a new generation of teachers who could support and collaborate with one another more effectively in professional communities. New application domains will outline new ways in environmental research, and new trends in education of our students.

4 Acknowledgment

The author is grateful for the support provided by the Institute for Environmental Studies in Prague. This work was supported in part by the Ministry of Education, Youth and Sports under grant: FRVS 131/2009A/b. The author acknowledges the support of a variety of other projects such as MSMT VS97/100, TACR TA01020428, GACR 526/09/0567, AVCR 1ET400760405, GACR 205/02/0898, MSM 0021620831, which partly supported presented environmental research case studies. The author wishes to thank the authorities of the City of Prague and its Institute of Municipal Informatics; Czech Office for Surveying, Mapping and Cadastre; and Czech Environmental Information Agency for providing the data of the living environment in the framework of student case studies.

5 References

[1] Boyd, D.S. and Foody, G.M. (2010) An overview of recent remote sensing and GIS based research in ecological informatics. *Ecological Informatics*, **6**, 25–36.

[2] Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. and Ghalsasi, A. (2010) Cloud computing-The business perspective. *Decision Support Systems*, **51**, 176–189.

[3] Sultan, N. (2009) Cloud computing for education: A new dawn? *International Journal of Informatics Management*, **30**, 109–116.

[4] Zeiler, M. (2010) *Modeling our World*. ESRI Press, Redlands.

[5] Allen, D.W. and Coffey, J.M. (2010) *GIS Tutorial: Advanced Workbook 3*. ESRI Press, Redlands.

[6] Mitchell, A. (2005) *GIS Analysis-Volume2: Spatial Measurements & Statistics*. ESRI Press, Redlands.

[7] Burrough, P.A. and McDonnell, R.A. (1998) *Principles of Geographical Information Systems*. Oxford University Press, Oxford.

[8] Johnston, C.A. (1998) *Geographic Information Systems in Ecology*. Blackwell, London.

[9] Fedra, K. (1999) Urban environmental management: monitoring, GIS, and modeling, *Computer, Environment and Urban Systems*, **23**, 443-457.

[10] Wade, T.G., Wickham, J.D., Zacarelli, N. and Riitters, K.H. (2009) A multi-scale method of mapping urban influence. *Environmental Modelling & Software*, **24**, 1252-1256.

[11] Maquire, D.J.. Batty, M. and Goodchild M.F. (2005) *GIS, Spatial Analysis and Modeling*. ESRI Press, Redlands.

[12] Goodchild, M.F., Steyaert, L.T., Parks, B.O., Johnston, C., Maidment, D., Crane, M. and Glendinning, S. (1996) *GIS and Environmental Modeling: Progress and Research Issues*. GIS World Books, Fort Collins.

[13] Holzbecher, E. (2007) *Environmental Modeling Using MATLAB*. Springer, Berlin.

[14] Matejicek, L. (2010) *Environmental Modeling with GPS*. Nova Science Publishers, New York.

[15] Clark, M.M. (1996) *Transport Modeling for Environmental Engineers and Scientists*. John Wiley & Sons, New York.

[16] Jørgensen S.E. and Bendoricchio, G. (2001) *Fundamentals of Ecological Modelling*. Elsevier, Amsterdam.

[17] Cromley, E.K. and McLafferty, S.L. (2002) *GIS and Public Health*. Guilford Press, New York.

[18] Costanza, R. and Jørgensen, S.E. (2007) *Understanding and Solving Environmental Problems in the 21st Century: Toward a new, integrated hard problem science*. Elsevier, Amsterdam.

[19] Dickson, K.L., Maki, A.W. and Cairns, J. (1982) *Modeling the Fate of Chemicals in the Aquatic Environment*. Ann Arbor Science, Michigan.

[20] Bies, D.A. and Hansen, C.H. (2010) *Engineering Noise Control*. Taylor & Francis, London.

[21] Pinder, G.F. (2002) *Groundwater Modeling Using Geographical Information Systems*. John Wiley & Sons, New York.

[22] Hannon, B and Ruth, M. *Modeling Dynamic Biological Systems*. Springer, New York.

[23] Seppelt, R. (2003) *Computer-Based Environmental Management*. Wiley-VCH, Weinheim.

[24] Argent, R.M., Perraud, J.M., Rahman, J.M. Grayson, R.B. and Podger, G.M. (2009) A new approach to water quality modeling and environmental decision. *Environmental Modelling & Software*, **24**, 809-818.

[25] Basnyat, P. Teeter, L.D. Lockaby, B.G.and Flynn, K.M. (2000) The use of remote sensing and GIS in watershed level analyses of non-point source pollution problems. *Forest Ecology and Management*, **128**, 65-73.

[26] Matejicek, L., Janour, Z. Benes, L., Bodnar, T. and Gulikova, E. (2008) Spatio-temporal modeling of dust transport over surface mining areas and neighbouring residential zones. *Sensors*, **8**, 3830-3847.

[27] Matejicek, L. and Kopackova, V. (2010) Changes in croplands as a result of large scale mining and the associated impact on food security studied using time-series Landsat images. *Remote Sensing*, **2**(6), 1463-1480.

[28] Tang, U.W. and Wang, Z.S. (2007) Influence of urban forms on traffic-induced noise and air pollution: Results from a modelling system. *Environmental Modelling & Software*, **22**, 1750-1764.

[29] Matejicek, L. (2010) Spatial Modeling of Air Pollution Based on Traffic Emissions in Urban Areas. In: Paterson, S.E. and Allan L.K. (eds) *Road Traffic: Safety, Modeling, and Impacts*, 521-534. Nova Science Publishers, New York.

[30] Cressie, N.A.C. (1999) *Statistics for spatial data*. John Wiley, New York.

[31] Johnston, K., ver Hoef, J.M., Krivoruchko, K. and Lucas, N. (2001) *Using ArcGIS Geostatistical Analyst*. ESRI Press, Redlands.

[32] Grizzetti, B. Bouraoui, F., Granlund, K., Rekolainen, S. and Bidoglio, G. (2005) Modelling diffuse emission and retention of nutrients in the Vantaanjoki watershed (Finland) using the SWAT model. *Ecological Modelling*, **169**, 25-38.

[33] Meals, D.W., Cassell, E.A., Hughel, D., Wood, L., Jokela, W.E. and Parsons, R. (2008) Dynamic spatially explicit mass-balance modeling for targeted watershed phosphorus management I. Model development. *Agriculture, Ecosystems and Environment*, **127**, 189-200.

[34] He, C. (2003) Integration of geographic information systems and simulation model for watershed management. *Environmental Modelling & Software*, **24**, 809-813.

[35] Miller, S.N., Semmens, D.J., Goodrich, D.C., Hernandez, M., Miller, R.C., Kepner W.G. and Guertin, D.P. (2007) The automated geospatial watershed assessment tool. *Environmental Modelling & Software*, **22**, 365-377.

[36] Pullar, D. and Springer, D. (2000) Towards integrating GIS and catchment models. *Environmental Modelling & Software*, **15**, 451-459.

[37] Neitsch, S.L., Arnold, J.G., Kiniry, J.R., Williams J.R. and King, K.W. (2002) *Soil and Water Assessment Tool: Theoretical Documentation*. Water Resources Institute, Texas.

[38] Matejicek, L. and Konradova, J. (2010) Spatio-Temporal Modeling by Field Computers with GPS. In: Matejicek, L. (ed) *Environmental Modeling with GPS*, 161-178. Nova Science Publishers, New York.

[39] Pires, A., Matinho, G. and Chang, N. B. (2011) Solid Waste management in European countries: A review of systems analysis techniques. *Journal of Environmental Management*, **92**, 1033-1050.

[40] Matejicek, L. (2008) Spatial Modeling and Optimization of Municipal Solid Waste Collection in Urban Regions. In: Golush, T.V. (ed) *Waste Management Research Trends*, 353-365. Nova Science Publishers, New York.

[41] Didion, M., Kupferschmid, A.D., Lexer, M.J., Rammer, W., Seidl, R. and Bugmann, H. (2009) Potentials and limitations of using large-scale forest inventory data for evaluating forest succession models. *Ecological Modelling*, **220**, 133-147.

[42] Matejicek, L. and Vavrova, E. (2010) Spatio-Temporal Modeling by Field Computers with GPS. In: Matejicek, L. (ed) *Environmental Modeling with GPS*, 145-159. Nova Science Publishers, New York.

[43] Purves, R.S., Medyckyj-Scott, D.J. and Mackaness, W.A. (2005) The e-MapScholar project-an example of interoperability in GIScience education. *Computers & Geosciences*, **31**, 189–198.

[44] Grunwald, S., Ramasundaram, V., Bruland, G.L. and Jesseman, D.K. (2007) Expanding distance education in the spatial science through virtual learning entities and a virtual GIS computer laboratory. *Journal of Distance Education Technologies*, **5**, 54–69.

[45] Yang, C., Raskin, R., Goodchild, M. and Gahegan, M. (2010) Geospatial cyberinfrastructure: past, present and future. *Computers, Environment and Urban Systems*, **34**, 264–277.

Teaching GUI Design and Event Handling Using Patterns

Hans Dulimarta and Jagadeesh Nandigam
School of Computing and Information Systems
Grand Valley State University
Allendale, Michigan 49401, USA
dulimarh@cis.gvsu.edu, nandigaj@gvsu.edu

Abstract - *Graphical User Interface (GUI) has become a common way to build and deploy desk-top software in introductory computer science courses. Students definitely enjoy building GUI programs as opposed to command-line programs. Writing a GUI program requires a paradigm shift from sequential processing to event-driven processing. Building a GUI program involves two main activities – design of user interface and implementing appropriate event handling. There are several approaches (code patterns or styles) to building GUI programs. Introductory computer science textbooks suitable for use in CS1 and CS2 courses tend to describe and use exclusively one approach to building GUI programs. This exclusive use of one approach to teaching GUI design is not effective as it can cause confusion when students move from one course to another and expect to see only the style of GUI program they have learned before. It is important that students are progressively exposed to different styles of GUI design independent of the approach suggested and used in the textbook adopted for the course. This paper presents various approaches to GUI design and event handling using Java Swing framework along with pros and cons of each approach.*

Keywords: GUI design, view patterns, event handling patterns.

1 Introduction

Since the 1980s Graphical User Interface (GUI) has been used as a standard means to deploy software. Certainly, some applications are designed to be “faceless” and they run without interacting with a human user. But, as future software developers, students need to be equipped with the necessary skills to build either type using proper techniques.

Writing a GUI program requires a paradigm shift from sequential processing to event-based processing. Callback functions must be provided to respond to various events generated by the GUI framework. There are several code patterns that can be employed when building GUI programs. Many textbooks [1, 2, 3] on Java programming used in CS1 and CS2 courses tend to describe one way of building a GUI program and all the GUI examples shown in the book are solved using this approach. When students move from one course to next (say CS1 to CS2), the book used in the next course may use a different style of GUI programming. This change in approach to GUI design from one book to another confuses students, especially those in the introductory computer science courses. Even the instructors teaching these courses must be aware of different approaches to building GUI programs when adopting a different textbook for a course. In these cases, the instructor must make a conscientious attempt to use the same approach as used in the textbook in an effort not to confuse students. It is important that instructors discuss and demonstrate with examples different approaches to building GUI programs and appropriateness of each of these approaches. Depending on the complexity of a GUI program to be developed, students will then be able to select the right approach to use for the problem at hand.

This paper describes various code patterns for teaching GUI design and event handling techniques in introductory CS courses. The rest of this paper is organized as follows. The Model-View-Controller (MVC) design pattern and its variants are discussed in Section 2. Sections 3 and 4 describe various patterns to view design and event handling. Section 5 discusses techniques for adding the model component to a GUI application. Section 6 describes alternative GUI design tools. Recommended approaches to teaching GUI design and event handling in introductory CS courses are outlined in Section 7. Section 8 ends the paper with concluding remarks.

2 Design Patterns

In object-oriented development, a large library of design patterns is used to assist programmers to discover inter-object relationships and structures commonly used in object-oriented programs. The book by the “Gang of Four” [4] is the *de facto* standard reference for design patterns, but the materials presented in the book are aimed for more experienced and professional programmers. Nevertheless, early CS students can benefit from understanding some fundamental design patterns and use them in writing complex multi-object programs [5].

One of the fundamental design patterns encountered by early CS students is the Model-View-Controller (MVC) design pattern, which also has inspired the implementation of a large number of Graphical User Interface (GUI) frameworks. This pattern gained popularity after its major use for building user interfaces in Smalltalk during the late 1970s [4]. Despite its popularity and wide-spread use in many real software projects, Hansen and Fossum claim that the pattern receives inadequate attention and CS educators should emphasize the design decisions that guide the development of an application [6]. The MVC design pattern consists of three kinds of objects:

- The model manages the data used by the application.
- The view presents the model in a form that allows the user to interact with the data.
- The controller handles user actions on the view and initiates proper actions for updating the data.

There are also variants of the MVC design pattern. Two of these variants are called *model-delegate* and *model-view-presenter*. In most applications, the view and controller are coupled very tightly. The resulting design pattern is known as *model-delegate*. The GUI framework in Java Swing employs this pattern.

The MVC pattern seems to imply that the model update is initiated by user actions on the view. In some applications, the model may be updated by an external source and the view must be updated accordingly. Consider a case when the user initiates a request to download remote data into the model. To keep the model and the view in sync with each other, the view must be updated only after the remote data is downloaded in full. For this use case, the view update must be initiated from the model. A common technique to accomplish this mechanism is to use the *Observer* pattern between the view and the model. This variant is known as *model-view-presenter* and was proposed by Fowler [7].

Beyond using well-known patterns, Bishop and Horspool [8] have identified common principles in GUI programming and use them in developing an XML-based GUI description language.

Using a simplified pattern than the standard MVC, Swing combines the view and controller roles into the *UI Delegate*. Different patterns to implement the view of a GUI application are presented in the next section. Section 4 describes several patterns to implement event handling in a GUI application.

3 View Patterns

In this section we will explore a number of techniques for implementing a GUI application using the Java Swing framework. We specifically exclude techniques used by modern IDEs that provide a visual GUI editor and generate the GUI code for Java developers. Instead, we will focus on techniques that can be used when the GUI code is hand-written by the users themselves.

Swing provides three top-level container classes: *JFrame*, *JApplet*, and *JDialog* as well as one general-purpose container class *JPanel*. Out of the three top-level container classes, we will focus our discussion on JFrame. With BorderLayout as its default layout manager, JFrame is a proper choice as a top-level container. The five sections of BorderLayout allow GUI components to be organized into coherent groups.

In contrast to JFrame, a container created from JPanel has the FlowLayout as its default layout. A FlowLayout does not provide a natural mechanism to organize its contents into coherent groups.

There are mainly three patterns to implementing the view in a GUI application. The table below summarizes these patterns. Rest of this section describes each pattern further with a sample code along with pros and cons of applying that pattern.

Patterns to Implement View in GUI App	
Pattern 1	Inheritance from JPanel
Pattern 2	Inheritance from JFrame
Pattern 3	JFrame as instance variable(s)

3.1 Pattern 1 – Inheritance from JPanel

We inherit a class from JPanel and place an instance of this panel into a JFrame's content pane. A typical Java code that uses this pattern is shown below:

```
public class MainView extends JPanel {
   public MainView() {
      setLayout(____);
      add(___);
      add(___);
   }
}

public class Main {
   public static void main(String[]
                                   args) {
      JFrame top = new JFrame();
      top.getContentPane().add(new
                              MainView());
      top.pack();
      top.setVisible(true);
   }
}
```

The above two classes are typically written as two separate files: MainView.java and Main.java. When more views are needed in the GUI, additional classes like "MainView" must be provided.

Pros: In general, this technique works well for a GUI program with a small number of widgets where they can be contained in a single panel.

Cons: When this technique is used, the event handling logic is usually written within the same class of the corresponding view panel. This approach may not facilitate easy updating to components in one view panel due to user actions in a different view panel. For instance, a text field in a panel in the north section of a JFrame is to be enabled/disabled based on the user action on a radio button in a panel in the east section of the JFrame. To implement such a feature, appropriate arguments must be passed from one view panel to another.

3.2 Pattern 2 – Inheritance from JFrame

For a more complex GUI design, a different technique should be used. To avoid interdependency across view panels as described earlier, we can move away from JPanel-based design and pull everything into a big JFrame. A typical Java code that uses this approach is given below:

```
public class GUI extends JFrame {

   public GUI() {
     add(viewOne(),BorderLayout.NORTH);
     add(viewTwo(), BorderLayout.CENTER);
     pack();
     setVisible(true);
   }

   private JPanel viewOne() {
      JPanel myView = new JPanel();
      myView.setLayout(____);
      /* setup components of this view */
      return myView;
   }

   private JPanel viewTwo() {
      JPanel myView = new JPanel();
      myView.setLayout(____);
      /* setup components of this view */
      return myView;
   }

   public static void main(String[]
                                args) {
      GUI mainwin = new GUI();
   }
}
```

Pros: This “all-in-one” approach may seem natural to early CS1 students when they are not exposed to “functional decomposition” yet. All GUI components (panels, buttons, text fields, menu items, etc.) needed in the program are declared as instance variables in the class so they can be accessed from anywhere in the class. Visual updates that require information from several components can be easily written.

Cons: The amount of code for designing the GUI using this pattern can be overwhelmingly long and difficult to debug. In order to remedy this, the code can be organized into several private methods, each responsible for managing one coherent group. Essentially, this is similar to the JPanel-based technique described earlier except any cross views dependencies can be easily handled.

3.3 Pattern 3 – JFrame as instance variable(s)

An alternative to the second technique, the GUI class can be designed without inheritance. Instead, a JFrame instance variable is defined in the GUI class.

```
public class GUI {
    private JFrame top;

    public GUI() {
       top = new JFrame();
       top.add(_____);
       top.add(_____);
       top.pack();
       top.setVisible(true);
    }

    /* private methods defined here */

    public static void main(String[]
                                args) {
       GUI mainwin = new GUI();
    }
}
```

Pros: This technique is appropriate for early CS1 students when they are not yet exposed to inheritance. In fact this technique is the most flexible technique of all, since it allows provision of multiple top-level containers, in which case several JFrame (or JDialog) instance variables can be defined.

4 Event Handling Patterns

One of the central facilities in a GUI framework is handling of events via proper callback functions. Swing implements this facility through events and event listeners. An event-handling object can be instantiated either from either a Listener or an Adapter.

The most common technique for event handling is to attach the listener (or adapter) to the corresponding view. This is usually accomplished by one of the following techniques or patterns. This can be accomplished by using one of the patterns shown in the table below. Each of these patterns are discussed in the following sections using the MouseMotionListener (or MouseMotionAdapter) as an example.

Patterns to Implement Event Handling in GUI App	
Pattern 1	View implements Listener
Pattern 2	Inner Classes for Listener
Pattern 3	Anonymous Classes
Pattern 4	Adapter Classes
Pattern 5	External Classes

4.1 Pattern 1 – View Implements Listener

The view class implements one or more listeners. A typical java code that uses this technique is shown below:

```
public class GUIView implements
                     MouseMotionListener {
   private JPanel landscape;

   public GUIView() {
      landscape = new JPanel();
```

```
      landscape.addMouseMotionListener(
                                    this);
   }

   public void mouseMoved(MouseEvent
                               m_event) {
   }

   public void mouseDragged(MouseEvent
                               m_event) {
   }
}
```

Pros: This technique works very well for a simple GUI program with a small number of widgets.

Cons: 1) Menus (JMenu & JMenuItem), buttons (JButton), and input fields (JTextField) are among the most popular components used by early CS students. In Swing, these three groups of components trigger ActionEvents that must be handled by ActionListeners. Using the above technique, the corresponding actionPerformed method may be overwhelmingly too long to write. 2) For a more complicated GUI program, developers may prefer to organize the mouse-handling logic into several methods. For instance, dragging the mouse over a 2D graphics may be handled differently from dragging the mouse over landscape of 3D objects. The above technique forces the developer to write everything into one big mouseDragged method.

4.2 Pattern 2 – Inner Classes for Listener

The second technique calls for declaration of private inner classes. Each class typically implements a specific Swing listener.

```
public class GUIView {
   private JPanel landscape;
   Handler2D motionHandler;

   public GUIView() {
      landscape = new JPanel();
      motionHandler = new Handler2D();
      landscape.addMouseMotionListener(
                          motionHandler);
   }

   private class Handler2D implements
                 MouseMotionListener {
      public void mouseMoved(MouseEvent
                          m_event) {
      }

      public void mouseDragged(MouseEvent
                          m_event) {
      }
   }
}
```

Pros: Event handling logic can be written into independent cohesive classes. Organization of event handling logic into separate (inner) classes facilitates easier refactorization.

Cons: This technique requires more code writing and including inner classes in a Java class may confuse some students who cannot clearly distinguish classes from objects.

4.3 Pattern 3 – Anonymous Classes

Since the inner classes are usually private, their visibility is limited to the GUIView itself, and the declaration and instantiation of Handler2D in the above technique can be merged into one step, by instantiating an object from an anonymous class.

```
public class GUIView {
   private JPanel landscape;

   public GUIView() {
      landscape = new JPanel();
      landscape.addMouseMotionListener(
                          motionHandler);
   }

   private MouseMotionListener
                 motionHandler =
      new MouseMotionListener() {
        public void mouseMoved(MouseEvent
                             m_event) {
        }

       public void mouseDragged(MouseEvent
                             m_event) {
        }
      };
}
```

Pros: Slightly more concise than the previous technique of using private inner classes.

Cons: The syntax for writing anonymous classes may seem too complex to comprehend by early CS students, especially when the IDE used does not provide auto completion / code assist feature that supports this syntax.

4.4 Pattern 4 – Adapter Classes

For Listeners that define more than one method, Swing provides a corresponding adapter. Adapters avoid writing empty callback methods.

Cons: A minor misspelling in writing the method names will result in the program failure to override the intended callback method. Most students may easily overlook this mistake, especially when they are under crunch time. This can be avoided by encouraging students to first use Listeners and later refactor the design to use Adapter in place of Listeners.

4.5 Pattern 5 – External Classes

Technically it is possible to write the event handling class as an external class. Using the four techniques described earlier, the event-handling classes (and methods) have direct visibility of all the instance variables declared by the View class. On the contrary, an external class does not have direct visibility, and hence any variables needed by the event-handler must be passed from the View class.

5 Adding the "Model" Component

So far, we excluded discussions on including the model into the picture. There seem to be only two options: to separate or not to separate. Some instructors may decide not to enforce model separation in the first few assignments. As students learn more about "functional decomposition," it makes sense to enforce model separation for later assignments. When model separation is enforced, the interaction between the model and the view components fits into one of the following two techniques:

5.1 Pull by View

Using this technique the View invokes two methods in the Model. The first method invocation triggers the model to update its internal state and the second method invocation to pull the new state and used for updating the view. This technique may seem easier to beginners and suitable when the updated state can be determined immediately. When updating the model's internal state takes too long to complete, the View may stay "frozen" and unresponsive for a noticeable duration.

5.2 Push by Model

Using this technique, the View makes only one invocation to trigger the model to update its internal state. The model will then push the new state to the view. Implemented improperly, this technique may entice students to write unnecessarily complex interaction between the two components. A better approach of properly pushing the model internal states to the View is to use the Observer and Observable pair.

6 Alternative GUI Design Tools

Many modern IDEs provide a feature for designing the user interface of a program using a drag-n-drop technique. This feature certainly saves developer time to create the GUI, but most of the time "hand-written" GUI code is much leaner than those generated by these IDEs. Using these tools, the event-handling code snippets can be injected into various sections of the program.

Some standalone GUI editors/designer tools may support multiple GUI platforms, and these tools usually allow the GUI design to be saved into a file (such as an XML file) for later retrieval. The source code for specific GUI platforms can be generated and compiled together with the rest of the program. A number of GUI frameworks like Gtk or the Android View system include facilities to "inflate" or "render" the visuals from an XML description.

7 Recommendations

The two sets of choices described above allow instructors to choose the best option, out of sixteen (or twenty) different combinations, that works well for a particular audience. However, since learning revolves as a progression, the baseline for selecting the "best" option keeps shifting as students learn more towards the end of the semester.

The available options as described above seem to share a common theme: choosing between inheritance and composition. We claim that there is no particular one option that fits best for all situations. After completion of a typical CS1 and CS2 sequence, it is important that students gain good understanding of the various patterns described here. More importantly, they are equipped with necessary knowledge to wisely choose the right pattern to apply in a project.

In our discussion above, we mentioned a number of pros and cons for each pattern or option. The drawbacks of one option can be carefully posed to be an opportunity for students to grow in their learning experience.

8 Conclusions

Building GUI programs require a paradigm shift from sequential processing to event-based processing. Two main activities in building a GUI program are design of user interface and implementation of event handling. This paper presented several approaches (as code patterns) to user interface design and event handling. It is important that students be exposed to different approaches to GUI design and event handling along with pros and cons of these approaches in CS1 and CS2 courses.

9 References

[1] David Barnes, Michael Kolling. *Objects First with Java*. 4th. Prentice Hall, 2009.

[2] Julie Anderson and Herve Franceschi, *Java Illuminated*. 3rd edition. Jones-Bartlett Learning, 2011.

[3] J Lewis and W Loftus, *Java Software Solutions*, 6th edition. Addison-Wesley, 2008.

[4] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns, Addison-Wesley, 1995.

[5] Prasun Dewan, "Teaching Inter-Object Design Patterns to Freshmen", *SIGCSE '05*, St. Louis, Missouri, 2005.

[6] Stuart Hansen and Timothy Fossum, "Refactoring Model-View-Controller", *CCSC Midwest*, Decatur, Illinois, 2005.

[7] Martin Fowler, GUI Architectures, http://www.martinfowler.com/eaaDev/uiArchs.html (accessed Mar 7, 2011).

[8] Judith Bishop and Nigel Horspool, "Developing Principles of GUI Programming Using Views", *SIGCSE '04*, Norfolk, Virginia, 2004.

CS Undergraduates' Perceptions of Self-Tests in PowerPoint: Can learning be enhanced?

P. A. Walcott
Department of Computer Science, Mathematics and Physics
The University of the West Indies, Cave Hill Campus
Barbados

Abstract - *PowerPoint is a ubiquitous multimedia authoring tool that is widely used in education for in-class presentations and the distribution of course materials; however a common complaint is that it is linear and non-interactive thus does not enhance learning. Although the research literature describes methods of improving interactivity in PowerPoint such as the use of quizzes, there is a dearth of studies on the effectiveness of these methods. In an attempt to help to fill this gap this paper describes a case study of three undergraduate computer science classes, over three semesters, which utilised self-tests that were integrated into their PowerPoint course materials. This study utilised self-reported questionnaires to analyse students' perceptions of the tests to determine whether learning was enhanced. Many of the participants stated that their recall, understanding and retention were enhanced. These results are important since they help educators to understand and plan how to use PowerPoint more effectively for teaching and learning.*

Keywords: self evaluation, self assessment, self tests, formative assessment, PowerPoint, enhancing learning.

1 Introduction

PowerPoint is a ubiquitous multimedia authoring tool that is considered one of the most widely used software applications [9]; and e-learning applications [4]. This tool is often used in education for classroom teaching and for the distribution of course material; however, there is an ongoing debate on the use and misuse of PowerPoint [17][9][22][21][8][10][15][6]. The use of quizzes in PowerPoint to improve interactivity has been proposed by several authors [3][12][13][14][1][7][19][20][11] however the effectiveness of these methods has not been thoroughly investigated [18].

Tufte [22] describes PowerPoint as evil and suggests that its use in education is inappropriate since children are not "learning to write a report using sentences, [rather] children are being taught how to formulate client pitches and infomercials." In the same vein Parker [17] states that PowerPoint "edits ideas" and has "an opinion … about the way we should think." He adds that people are communicating with "no paragraphs, no pronouns" and that this is even happening in university lectures. Thompson [21] asks whether PowerPoint makes us stupider citing the example of NASA's Investigation Board who concluded that one of the causes of a space shuttle crash was a confusing PowerPoint slide; and McDonald [15] argues that creating PowerPoint presentations does not improve communication skills, creativity and critical thinking.

Clearly, there are some strong views about the way PowerPoint is being misused, however it is important to present a balanced argument. Keefe & Willett [8] lists three compelling reasons why PowerPoint should be used in the classroom:

> *… its [PowerPoint] suitability as a powerful and easily learned authoring system for course materials; its ubiquitous availability to students, courtesy of the free Microsoft PowerPoint viewer; and its capability of coexisting with an overall course management environment.*

Essentially PowerPoint can serve as a ubiquitous course authoring system that links to external resources (in a non-linear manner) such as discussion forums and journal articles [8]. It can also help students identify main ideas and their supporting details, an important reading and comprehension skill [10].

In response to Tufte's comments about PowerPoint being evil, Doumont [6] argues that Tufte does not differentiate between oral and written communication (the purpose of a document is to convey detailed evidence at a high transfer rate of words and data, while oral presentations are much slower and richer) thus his conclusions are flawed. Doumont [6] did agree however that many slides are poorly designed especially those used to help the speaker "remember what to say (or, worse, to read to the audience)."

The argument that PowerPoint is linear and non-interactive is based on the way many course designers use the tool, rather than its capabilities. Studying methods of improving teaching and learning in PowerPoint is therefore justified. One possible strategy is to integrate self-assessment (self-tests) into PowerPoint teaching materials [18]. Self-assessment used in this way can helps students determine those topics they understand and those that they do not; this is essentially formative assessment.

Formative assessment is the "diagnostic use of assessment to provide feedback to teachers and students over the course of instruction [2]." It is important to student success because it informs the teacher about the student's progress, identifying trouble areas and allowing adjustments in instructional practice [2]. However, formative assessment is not only useful for teachers but can also be used to help students identify gaps in their knowledge through self-evaluation [2]. This is especially vital for online learners who are expected to become independent learners.

Cliff, Freeman, Hansen, Kibble, Peat, Wenderoth [5] further clarifies the importance of formative assessment by stating that:

> *Formative assessment aims to inform learners about their progress toward achieving the goals and objectives of a program. It is intended to foster learning, helping students to develop under conditions that are non-threatening.*

When formative assessment is placed online it has the added advantage of allowing students to become independent and self-directed [16]. One type of formative assessment that has been used with some success is online quizzes [5]. These online assessments when blended with face-to-face teaching are "leading to more meaningful learning and conceptual understanding [5]."

There are three main methods of adding assessment / quizzes (self-tests) to PowerPoint reported in the literature. The first of these methods uses action buttons and hyperlinking [3][19][20][1]. The second uses Microsoft Visual Basic for Applications (VBA) [7][12][13][14], and the third utilises third party software such as Impactica [11].

Although the literature provides us with an in-depth commentary on the use and misuse of PowerPoint, and how it can be used for self-assessment, it rarely reports on how the use of self-assessment in PowerPoint may enhance learning. This clearly represents an opportunity for the study reported in this paper to contribute to the existing literature.

1.1 Overview

The remainder of this paper is organized as follows. In section 2 the purpose of the study and research questions are discussed. In Section 3, the contextual framework is presented, followed by the method in section 4. In section 5 the results are presented and in section 6 the discussion. Finally, the conclusions are presented in section 7.

2 Purpose of the Study and Research Questions

The purpose of this study was to determine whether self-tests in PowerPoint could enhance learning in undergraduate CS classes. In order to determine this, a case study addressing the following research questions was conducted:

- How many self-tests did students run and how often?
- Can self-test in PowerPoint help students to reinforce learning and understanding of the subject matter?
- Can self-test in PowerPoint help students retain course materials more easily?
- Can self-test in PowerPoint help students recall course materials more easily?

3 Contextual Framework

This study was conducted at the University of the West Indies (UWI), Cave Hill Campus in Barbados which is the smallest of the three UWI campuses; the other two campuses being located in the Caribbean islands of Trinidad and Jamaica. At the time of the study the campus comprised of five Faculties, namely the Faculty of Law, the Faculty of Pure and Applied Sciences, the Faculty of Humanities and Education, the Faculty of Social Sciences and the Faculty of Medicine. It offered both undergraduate and post-graduate programmes in law, social sciences, the arts, education, medicine and the sciences and had a student enrolment of approximately 8,500 including undergraduate and post-graduates.

In the Faculty of Pure and Applied Sciences was the Department of Computer Science, Mathematics and Physics which offered the B.Sc. in Computer Science and B.Sc. in Information Technology degrees. Students in this department were also able to undertake joint majors such as Computer Science and Accounting and Computer Science and Management. Electives offered in these programmes included Electronic Commerce and Information Systems which were both 4-credit courses.

Both the Electronic Commerce and Information Systems courses were offered in a blended environment where either five or six hours of face-to-face instruction were provided. Typically students were expected to attend, over a twelve week period, three hours of lectures, two hours of laboratories and (in some cases) a one hour tutorial. Students were able to download course materials from the course website and engage in online discussions.

4 Method

This case study utilised a self-reported questionnaire for data collection. The questionnaire was administered by the author, who was the instructor of the courses; the sample was therefore convenient.

4.1 Participants

Thirty-six computer science students participated in this study. These students were drawn from two Electronic Commerce classes (Fall 2009 and 2010) and one Information Systems class (Spring 2010); thirteen students belonged to the Fall 2009 Electronic Commerce class, ten were from the Spring 2010 Information Systems class and thirteen from the Fall 2010 Electronic Commerce class.

4.2 Data collection procedures

A pilot study was conducted using ten students from the Spring 2009 Information Systems class in order to test the suitability and design of the survey instrument.

In the main data collection phase students were informed about the purpose of the study and were assured that the results would remain anonymous and confidential. These surveys were administered at the beginning of the final session of each course and collected immediately after their completion. Thirty-eight questionnaires were distributed and all were completed and returned. Five of the questionnaires were subsequently removed; two belonging to students how had completed the survey in two different classes; and three because they did not use the quiz facility at all.

4.3 Survey Instrument

The survey instrument was a questionnaire which inquired the students' demographic and academic information, namely their email address, gender, age, major, and level of study. It also inquired whether their owned a personal computer and the type of Internet access that they had. It sought to determine how many of the self-tests were used and whether these self-tests: reinforced learning; helped the student to retain and recall course information easily; forced the student to reflect on the subject matter; motivated the student; and helped them to engage in deep learning.

The instrument also questioned the number of times the self-tests were run; whether the peek facility (this allowed students to peek at the slide in the course materials which provided them with information that should help them answer the question) was used, and if it was used then how often and whether it improved the comprehension of the material. Students were also asked whether they preferred immediate or non-immediate feedback from the tests.

General questions about the self-tests were also queried, namely: whether the self-tests were used for test preparation; what were the benefits of the self-tests; were they easy to use; whether they made learning fun; and whether the students felt more comfortable doing the test knowing that the results were not fed back to the instructor. Additional questions were asked about: the average download times of the materials; the ease of installation of the teaching materials; the security of their computers when using the teaching materials; and whether students had any improvement suggestions for the teaching materials.

4.4 Design of the course materials

The course materials for the Electronic Commerce and Information System courses were distributed in PowerPoint format. For each session (a week of instruction) a PowerPoint file would be posted online containing a list of session objectives, the session content, hyperlinks to external resources, a self-test of at least ten (multiple choice, true/false or fill-in-the blank type) questions, a report card, a list of references and in some cases a reflective question.

The students were allowed to choose whether they wanted to complete a given self-test in immediate feedback mode or non-immediate feedback mode which simply meant that the user would be told whether they answered the question correctly immediately after answering the question (immediate) or not (non-immediate). If a student had difficulty answering a given question he/she could choose to "peek", that is jump to the content slide which helped him/her to answer the question. Finally, when the test was completed the student could view his/her report card which provided the final score as a percentage, and for each question indicating whether the student got it correct or incorrect.

Each of these PowerPoint files were first converted to a PowerPoint show (to run immediately upon opening) and then compressed into a .zip file in order to reduce download times. Examples of these course materials may be found at http://www.cavehill.uwi.edu/staff/eportfolios/paulwalcott/notes.html.

4.5 Limitations

The main limitations of the study were that the computer science courses used were conveniently selected, and the survey instrument was administered at the end of the semester but before the final examinations. Note that the reported results are therefore based on course work assignments and course tests, as well as students' intended use of the tool for studying for the final examination.

5 Results

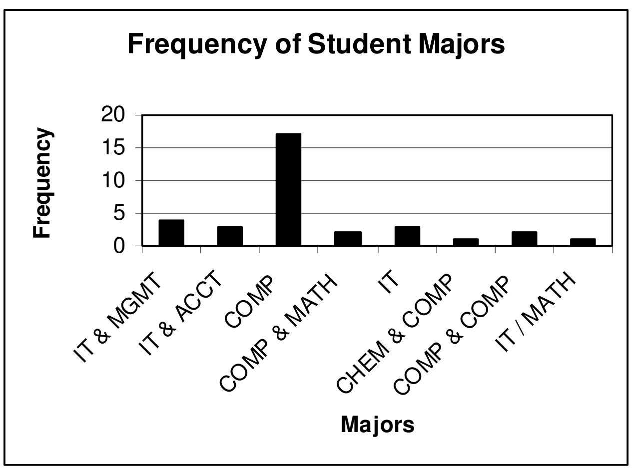
The results of this study are presented as summative and descriptive.

5.1 Demographics and academic background of students

The results presented in this section are based on the survey instruments received from thirty-three participants, twelve females and twenty-one males. All of these students were in their final year of study. Their ages ranged from 19 years to 34 years old, with an average age of 24.1 years.

Eight different majors (including joint majors) were being undertaken by these students (Figure 1). These majors were: Information Technology and Management (IT & MGMT); Information Technology and Accounts (IT & ACCT); Computer Science (COMP); Computer Science and Mathematics (COMP & MATH); Information Technology (IT); Chemistry and Computer Science (CHEM & COMP); a double major in Computer Science (COMP & COMP) and Information Technology with Mathematics (IT / MATH).

Figure 1. The frequencies of students' majors



5.2 How many self-tests did students run and how often?

The number of self-tests ran varied from class to class, however on average most of them were utilised by the students. Alternatively, the average number of times each students ran a given self-test was 2.6.

5.3 Can self-test in PowerPoint help students to reinforce learning and understanding of the subject matter?

Of the thirty-two students who answered this question (one student did not respond), approximately 84 percent said that the self-tests helped them to reinforce their learning and understanding of the subject matter. The other 16 percent stated that the self-tests helped them only sometimes. One student found that the self-tests "… enables the user to think and understand the focus of the session." Another believed that they "show[ed] points to focus on." Even more importantly, some students believed that it helped to summarise information; "It helped in giving a summary and understanding [of] specific points," and "… the quizzes help to summarise the current session notes."

The self-tests also motivated students to re-read the course notes until they had a complete understanding of the material. This was seen from the comments of some students who noted that they read the course materials and took the self-test iteratively until they got a "full score" on the self-test; one of these students actually said that the self-test motivated her to pay attention as she read the course materials, so that she could "gain the highest score possible."

Other students noted similar positive effects gained from the integrated self-tests including: it "help[ed] to brush up on objectives;" "in the event that I really did not know it was easy to use the peek function to get an idea of the answer;" "it allowed me to review information and have a better understanding of the subject matter;" and "it tests you as soon as you learn" thereby improving understanding.

5.4 Can self-test in PowerPoint help students retain course materials more easily?

A moderate number of students also believed that the self-tests helped then to retain the materials more easily. Approximately 67 percent of the students stated yes while another 12 percentage stated somewhat. Only 15 percent did not think that they were able to retain materials more easily.

One student stated that "the whole concept of instant feedback and peeking were extremely useful to the retention process." Another noted that "it was easier to retain the information after doing a quiz." This was also echoed by another student who said that "it [the quiz] helps [me] to retain the areas covered in the quiz."

"It [the quiz] served as a quick refresher and helped to cement the topics covered" was the impression of a student; while another went as far as comparing it to other courses stating that "in comparison to other courses that don't have quizzes, I seem to soak up the large amount of information faster."

5.5 Can self-test in PowerPoint help students recall course materials more easily?

Some 58 percent of the students believed that the quizzes helped them to recall the information more easily. Another 27 percent stated that it help them sometimes while 16 percent stated that it did not help them.

Several students found that the quizzes helped them to recall information. For example a student noted that the quizzes "help[ed] trigger memory." Another said that "yes they jogged my memory," and one stated that a quiz "helps me in recalling information relating to the question that is presented."

6 Discussion

Many students in the study said that the tests integrated into the teaching materials reinforced learning and improved retention and recall. In addition, the interactive nature of the tests, including the peek facility and the immediate feedback, helped them to quickly determine what they understood and what they did not.

The self-tests motivated students to succeed; they fostered independent learning and helped students to perform self-evaluation. One student said that knowing that she was going to do the self-test made her pay closer attention while reading the materials while others re-read the material multiple times in order to gain better scores on the self-tests. Motivation and independent learning are highly desirable student characteristics especially in online or blended environments.

A further benefit of self-tests is the reduction in the amount of time the course instructor needs to spend providing feedback since feedback is provided by the tests. This benefit would certainly be more pronounced in larger class sizes.

The very high usage of self-tests noted in this study resulted from the accessibility and ease of use of the tests (since they were integrated into the course materials). Also, many students simply did not purchase the texts for these courses because they believed that CS information was readily available on the Internet and that course instructors would provide teaching materials online anyway. This high self-test usage could be replicated in other context if it is ensured that any PowerPoint materials distributed contain information that cannot easily be sourced elsewhere.

A very important characteristic of these self-tests is that they help students to gauge their understanding of the given topic. If after taking the tests they found that they could not answer the questions they would either review all of the materials, or use the peek facility to review the materials associated with the answers then repeat the test. The tests therefore helped them to figure out what they did not fully understand and allowed them to immediately address these issues.

7 Conclusions

Despite the ongoing debate regarding the usefulness of PowerPoint this research has shown that integrating self-tests into teaching materials created in PowerPoint can enhanced student recall and retention, and can reinforce learning.

It has been shown that self-tests, with immediate feedback and peeking in PowerPoint, can help students to become motivated and independent learners.

This is good news for course instructors since PowerPoint is widely used in academia and in some cases is the only available tool. With a little creativity and self-tests PowerPoint can provide a flexible learning environment for students. The method discussed in this paper of integrating self-tests into PowerPoint can be adopted by course instructors of any discipline and is especially useful for faculty of "cash strapped" institutions who cannot afford more expensive tools.

8 Acknowledgements

The author wishes to acknowledge and thank all of the students who participated in this study.

9 References

[1] Geetesh Bajaj. "Simple Quizzes in PowerPoint", Computer Companion, Retrieved February 1, 2009 from http://www.computorcompanion.com/LPMArticle.asp?ID=210.

[2] Carol Boston. "The concept of formative assessment"; Practical Assessment, Research & Evaluation, Vol. 8, Issue 9, 2002. Retrieved May 23, 2011 from http://PAREonline.net/getvn.asp?v=8&n=9.

[3] Terence Cavanaugh, Cathy Cavanaugh. "Interactive PowerPoint for Teachers and Students"; in C. Crawford et al. (Eds), Proceedings of Society for Information Technology and Teacher Education International

Conference 2000 (Chesapeake, VA AACE), 496-499, 2000.

[4] Bryan Chapman and the staff of Brandon Hall Research. "PowerPoint to E-Learning Development Tools: Comparative Analysis of 20 Leading Systems"; Brandon Hall Research, Sunnyvale, CA, 2006.

[5] William Cliff, Scott Freeman, Penelope A. Hansen, Jonathan D. Kibble, Mary Peat, Pat Wenderoth. "Is formative assessment an effective way to improve learning?"; Advances in Physiology Education (A symposium at Experimental Biology 2008), Vol. 32, 337-338, 2008.

[6] J Doumont. "The Cognitive Style of PowerPoint: Slides are Not Evil"; Technical Communications, Vol. 52, Issue 1, 64-70, 2005.

[7] Ellen Finkelstein, "Creative Techniques: easily create a quiz in PowerPoint using Visual Basic for Applications"; Presentations, Jan 2005.

[8] David Keefe, J. Willett. "PowerPoint as an authoring tool for science faculty"; Cell Biology in Education Journal, Vol. 3, Issue 3, 2004.

[9] Julia Keller. "Killing me Microsoftly with PowerPoint"; Chicago Tribune, Jan 2003. Retrieved May 23, 2011 from http://www.gbuwizards.com/files/chicago-tribune-julia-keller-05-january-2003.htm.

[10] Cynthia Lanius. "PowerPoint, Not Your Grandmother's Presentations, but is it Evil?"; Cell Biology Education, Vol. 3, Issue 3, 158-160, 2004.

[11] Alan Lawee. "Impatica for PowerPoint and Impatica OnCue -- Empowering content specialists"; in M. Driscoll & T. Reeves (Eds.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2002 (Chesapeake, VA: AACE), 1758-1760, 2002.

[12] David Marcovitz. "Multimedia in the Classroom With PowerPoint and VBA"; in C. Crawford et al. (Eds.), Proceedings of Society for Information Technology and Teacher Education International Conference 2001 (Chesapeake, VA: AACE), 941-946, 2001.

[13] David Marcovitz. "Making PowerPoint Powerful Using VBA To Add Interactivity"; in C. Crawford et al. (Eds), Proceedings of Society for Information Technology and Teacher Education International Conference 2003 (Chesapeake, VA AACE), 2765-2767, 2003.

[14] David Marcovitz. "Powerful PowerPoint for Educators: Using Visual Basic for Application to Make PowerPoint Interactive". Libraries Unlimited, Westport, CT, 2004

[15] Kim McDonald. "Examining PowerPointlessness"; Cell Biology Education, Vol. 3, Issue 3, 160-161, 2004.

[16] Meg O'Reilly, Karey Patterson. "Assessing learners through the WWW"; in H Ashman & P Thistlewaite (eds), Proceedings of the Seventh International World Wide Web Conference , Brisbane, Qld., 1-7 April, Computer Networks and ISDN Systems, Vol. 30, Issue 1-7, 727-729, 1998.

[17] Ian Parker. "Absolute PowerPoint: Can a software package edit our thoughts?"; The New Yorker, May 28, 76, 2001

[18] Lejla Rovcanin, E. F. M. O'Loughlin. "Creating and Managing learning objects with common office tools"; 2004. Retrieved January 1, 2011 from http://staffweb.itsligo.ie/staff/bmulligan/EdTech2004/papers/rovcaninloughlin.pdf

[19] Pavel Samsonov. "Creating quizzes and clickable maps with PowerPoint"; in E. Pearson & P. Bohman (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006 (Chesapeake, VA: AACE), 1437-1438, 2006.

[20] Pavel Samsonov. "Simple Techniques to Create Interactive Reviews in PowerPoint"; in G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2007 (Chesapeake, VA: AACE), 1826-1834, 2007.

[21] Clive Thompson. "PowerPoint makes you dumb"; New York Times, December 14. 2003. Retrieved January 1, 2011 from http://www.nytimes.com/2003/12/14/magazine/14POWER.html

[22] Edward Tufte. "Power Point is Evil"; Wired Magazine Online, 2003. Retrieved December 20, 2009 from http://www.wired.com/wired/archive/11.09/ppt2_pr.html.

MathFax: An Online Adaptation of the Elementary Learning Game "Math Facts"

Nicholas Handelman and Donald R. Schwartz
Department of Computer Science
Millsaps College
Jackson, MS USA

Abstract - *In the game "Math Facts", students are given up to sixty seconds to answer up to twenty elementary math problems. MathFax is an online adaptation of that game. Proposed by a local elementary school and developed by students in our Software Engineering class, MathFax goes far beyond the rudimentary paper and pencil version. It was developed for use on a WAMP server and offers many features: creating users, classes and tests and viewing test results. The paper describes these features in depth, details the database and discusses the current status of and future work for MathFax.*

Keywords: Software Engineering, Service Learning, Group Projects, Elementary Education, Innovations in Education

1 Introduction

For many of us, the days of elementary school are a distant memory. It was a time when our most pressing concern was "when is recess", not "when is my next deadline". Well, MathFax should serve as a reminder of that time. This software is an online adaptation of the elementary learning game often entitled "Math Facts" or "Math Fax." For those to whom this is unfamiliar, students are given a sheet containing twenty elementary math problems, consisting of addition, subtraction, multiplication, division or any combination thereof. The answer to any given problem is a non-negative integer, generally no greater than the low hundreds. Students are given sixty seconds to answer as many of these problems as they can, with the goal of improving their problem-solving skills over time.

MathFax began as one of several service-learning projects undertaken by our Software Engineering class, which was comprised of ten undergraduate students, during the Spring 2010 semester. Descriptions of service-learning projects and how these projects are managed can be found in [1][2][3][4][5][6][7]. The MathFax was proposed by St. Anthony's Catholic School, a local elementary school. The vast majority of the software was completed during that semester; however, a few students continued to refine the software after the semester ended.

This paper focuses on the software itself. Included are details on the chosen server and database: a WAMP server supporting Apache, PHP and MySQL. The bulk of the paper discusses the features which the software has to offer.

2 Server

For our development and hosting platform, we chose to use a WAMP (Windows Apache MySQL PhP) server. Using WAMP greatly eases the burden of installing and setting up the individual components. WAMP is an open source project created by Romain Bourdon and a team of PHP developers [8].

MathFax has been successfully tested on WampServer 2.1e with Apache, PHP and MySQL versions 2.2.17, 5.3.5 and 5.5.8, respectively. It has also been tested on WampServer 2.0 with Apache, PHP and MySQL versions 2.2.8, 5.2.6 and 5.0.51b, respectively.

3 Database

The database for our software consists of nine tables, each of which will be briefly described in this section.

3.1 User

This table contains information related to system access (login), and valid user permissions within the system. It contains four fields: 'userid', 'password', 'status' and 'privilege', which takes one of three values: admin, teacher or student.

3.2 Faculty

This table contains four fields: 'facultyid', 'first_name', 'mid_name' and 'last_name'. 'facultyid' uniquely identifies each teacher added to the system, and each 'facultyid' matches exactly one 'userid' in the user table.

3.3 Student

This table contains twelve fields: 'first_name', 'mid_name', 'last_name', 'gender', 'grade', 'time_goal' and six others. 'studentid' uniquely identifies each student added to the system, and each 'studentid' matches exactly one 'userid' in the user table. 'num_goal' holds the student's score goal (number correct within the time limit), while 'clock' holds the student's time goal (time taken if the student has reached the maximum 'num_goal'). 'ada' determines if the student's test screen background is white or yellow, while 'show_clock' determines if the student's on-screen test clock is visible or hidden. Finally, 'ClassID' matches the student to a unique class.

3.4 Class

This table contains seven fields: 'classname', 'grade', 'classyear', 'status' and three others. 'classid' is a unique identifier assigned to each created class. 'facultyid' identifies a unique teacher who teaches this class. 'classsem' is the semester the class begins, either fall or spring.

3.5 Test

This table contains seven fields: 'name', 'status' and five others. 'testid' is a unique identifier assigned to each created test. 'type' is a description of the test. 'style' indicates whether the test uses a vertical question format or a horizontal question format. 'grade' identifies the grade for which the test was created. 'time_limit' is the amount of time students are given to take the test.

3.6 Assigned_test

This table contains four fields: 'testid', 'classid', 'taken' and 'studentid'. The significance of this table will be discussed in section 4.6.3, Assigning tests and taking them.

3.7 Complete_test

This table contains four fields: 'testid', 'studentid', 'questionid' and 'answer'. The significance of this table will be discussed in section 4.6.3, Assigning tests and taking them.

3.8 Question

This table contains seven fields. 'questionid' is a unique identifier assigned to each created question. 'operand' is an operator (+, -, /, *). 'first_val' and 'sec_val' are the values upon which 'operand' operates. 'answer' is the answer associated with a this question. 'grade' identifies the grade for which the question was created. 'testid' indicates the test on which this question will appear.

3.9 Score

This table contains seven fields: 'studentid', 'testid', 'num_correct', 'num_goal', 'time_take', 'time_goal' and 'date_taken'. The significance of this table will be discussed in sections 4.6.3, Assigning tests and taking them and 4.7, Viewing test results.

4 Software

4.1 Compatible browsers

MathFax is fully compatible with the most up-to-date versions of Google's Chrome internet browser and Apple's Safari browser. In addition, Mozilla's Firefox browser is supported, though the layout may not always appear the same as it would in Chrome or Safari. Microsoft's Internet Explorer is not supported: the buttons do not function.

In each of these supported browsers, it is necessary to turn off the "AutoFill" feature. Since students taking a test must enter their answers in a text field, this browser feature must be turned off so the distracting "AutoFill" pop-ups do not disrupt the students' concentration.

4.2 Users and their privileges

There are three user types in MathFax: administrators, teachers and students. Each of these user groups has its own specific functions that form the whole of the software. For user support, three user's guides, one specific to each type of user, are included with the software and are accessible on-screen once a user has logged in. These guides may also be printed if necessary.

4.2.1 Administrators and their privileges

Administrators are responsible for setting up the system and maintaining the information contained therein. A default administrator account is provided with the system, but it is the primary administrator's responsibility to create new administrator(s) and delete the default account, for security purposes. Next, administrators have sole responsibility for adding teachers, classes and students to the system, and deleting those entries if necessary. They have access to all test results contained within the system and access to all three user's guides: administrator, teacher and student. When the end of the year comes around and students need to be promoted to the next grade and year, administrators have access to a feature that performs that function.

4.2.2 Teachers and their privileges

Teachers are responsible for testing and monitoring students' progress. First, an administrator must create accounts for them to access the system, assign them to classes and put students in those classes. Teachers then have the ability to create and administer tests, view all test results contained within the system and modify student specifics as necessary. In addition, they have access to two user's guides: teacher and student.

4.2.3 Students and their privileges

Students are the users for whom the system is designed, and they will comprise the majority of users. Once they are created, they will have access to the four following features: take an assigned test, take a practice test, view their own results and view their own score and time goals. Students have access to the student user's guide if they require assistance.

4.3 Logging and sessions

MathFax uses a fairly simple logging system. Someone wishing to access the software is required to enter a username and password. If the username and password match a user's respective information stored in the 'user' table, that person is granted access. If the information entered is incorrect, the user will remain on the login screen, and that information will be cleared from the screen.

If a login is successful, the following three actions will occur. First, the user's 'userid' will be stored as a session variable for use within the software. Next, the user's 'privilege' will be stored as a session variable for access purposes within the software. Finally, the user will be brought to the appropriate main menu, determined by his or her 'privilege'.

Sessions are terminated under only two conditions. If a user selects "logout" once he or she is logged in, the session will terminate. Also, if a user closes the browser while logged in, the session will terminate.

4.4 Main menu and side menu

When a user has successfully logged in, he or she is presented with a main menu. In addition, users are provided with a left-aligned side menu on all screens, except those where a student is taking a test. As shown in Figure 1, the side menu gives the user easier access to the different software functions, without having to return to the main menu. However, if the user wishes to return to the main menu (like the one shown in Figure 2), a "Home" button is provided in the upper left corner of every screen on which the side menu appears. The contents of both menus are determined by the user's privilege, assigned at login.

Figure 1: Side Menu (pictured on its side at a 69% zoom)

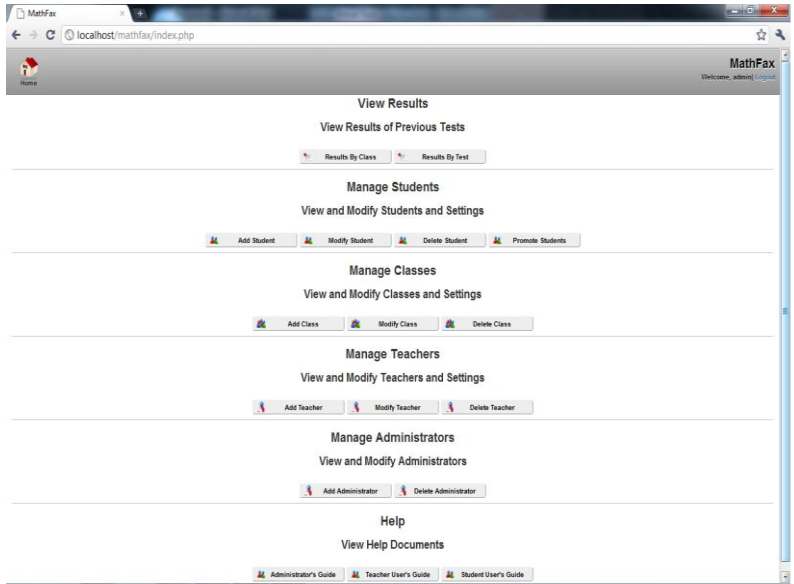


Figure 2: Administrator's Main Menu (pictured at 83% zoom)

4.5 Managing users and classes

This section discusses adding, modifying and deleting the following: administrators, teachers, students and classes. Only administrators have access to these features; however, teachers do have the ability to modify students as well.

A noteworthy feature of the software applies to user account creation. Since 'userid' is the primary key field of the 'user' table, no duplicates for this value can be stored in the table. However, multiple users may be assigned the same 'userid' value during account creation. To solve this problem, the software simply appends the next available digit to the desired name. For example, suppose there exists an account with a 'userid' of "NewUser". The next user to request the 'userid' of "NewUser" will be assigned the 'userid' of "NewUser1". The next one will be "NewUser2" and so on. The significance of this feature will be discussed in later in this section.

4.5.1 Managing administrators

Adding administrators is a simple process. The new administrator chooses his or her username and password and enters a confirmation password. The account is created and inserted into to the 'user' table.

Deleting administrators is also a simple process. A drop down box lists all administrators. One of these is chosen, and the account is deleted from the 'user' table. Administrators can delete themselves, though they will remain logged in. Also, the last remaining administrator cannot be deleted from the system, a feature that is hard-coded into the PHP.

4.5.2 Managing teachers

4.5.2.1 Adding teachers

When creating a teacher, five fields are requested. A first name and last name are required, but a middle name is optional. Also, a password must be entered and confirmed. Upon successful completion of all required fields, the new teacher's 'userid' is generated and added to the 'user' table. In Figure 3, the teacher "New School Teacher" is created, with a 'userid' of "NewTeacher".

Figure 3: Teacher Is Successfully Added

This teacher's 'facultyid' (same as 'userid') and first, middle and last names are inserted into the 'faculty' table.

4.5.2.2 Modifying teachers

The first action allows an administrator to search for the teacher by last name, as shown in Figure 4.

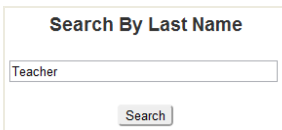


Figure 4: Search By Last Name

The software matches the entered phrase to any last name that contains that phrase and lists those names in a drop down box. If the search field is left blank, all teachers will be listed in the subsequent drop down box.

Once a teacher is selected, that teacher's first, middle and last names can be modified, though the 'userid' cannot be changed. Also, the password can be changed by entering a new password, confirming it and selecting "Yes" from the "EditPassword?" drop down box. The same fields required when creating a teacher are required here, though a password is not required if the "EditPassword?" box states "No". A confirmation screen appears if the modification is successful.

4.5.2.3 Deleting teachers

Deleting teachers works the same way as deleting administrators. A teacher is selected from a drop-down list and deleted from both the 'user' and 'faculty' tables. All teachers can be deleted.

4.5.3 Managing students

Students can be managed in the same way as teachers. There is also an option to "Promote Students."

4.5.3.1 Adding students

There are six requested fields. The four fields 'First Name', 'Last Name', 'Grade' and 'Class' are required. The fields 'Middle Name' and 'Gender' are optional. Upon successful completion of all required fields, the new student's 'userid' and 'password' are generated and added to the 'user' table. An example is shown in Figure 5.

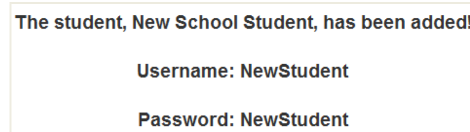


Figure 5: Student Is Successfully Added

Also, the new student's 'studentid' (same as 'userid') and other completed information will be inserted into the 'student' table. Note that a student can't be created until a class is created, since there will be no 'Class' option to select.

More options are available for students and are described in the following section.

4.5.3.2 Modifying students

First, a student is selected. This process is performed the same as it is for modifying teachers.

Once a student is selected, the same information that was used to create the student can be modified. The student's password can be modified here as well, using the same process as is used in modifying a teacher. All fields except for 'Middle Name' are required, though a password is not required if the "Edit Password?" drop down box is set to "No". Four new fields: 'Number Goal', 'Time Goal', 'Yellow Screen' and 'Show Clock', are presented.

A student's 'Number Goal' is the number of questions he or she is aiming to tie or exceed when taking an assigned test. It must be given a value zero through twenty, inclusive. A student's 'Time Goal' only applies if the student has a 'Number Goal' of twenty. Once a student has reached this level, the new overall goal becomes to finish the twenty questions in a shorter time, denoted by 'Time Goal'. It must be given a value one through sixty, inclusive.

The 'Yellow Screen' feature allows students to take their tests with a yellow, as opposed to white, background. Selecting "Yes" in the drop down box indicates that the student wishes to use a yellow background, while "No" indicates a white background. This value is stored in the 'ada' field in the 'student' table: 1 for yes, 0 for no. The 'Show Clock' feature allows students to hide the on-screen clock while taking tests. Selecting "Yes" in the drop down box displays the clock, while "No" hides the clock. This value is stored in the 'show_clock' field in the 'student' table: 1 for yes, 0 for no. It must be noted that both of these drop down boxes are set to "No" when the modification screen appears. They must be set to "Yes" every time a student is modified, if the student prefers it. An example of these options can be seen in Figure 8: Example Test Screen.

4.5.3.3 Deleting students

This feature is performed using the same process as is used when modifying a student. A search by last name is performed and then a student is selected from the subsequent drop down box. The selected student is deleted from both the 'user' and 'student' tables, and that student's assigned tests, completed tests and scores are deleted from their respective tables. All students can be deleted.

4.5.3.4 Promoting students

When the end of the year comes around and students will be moving to the subsequent grade and year, this feature allows for easy movement of all students from one class to another class. The process is as follows.

First, a current class is selected from a drop down box listing all classes. Next, one of two processes can occur. If the selected class is a sixth grade class, the administrator is presented the option to delete the students in that class along with the class. If the selected class is a first through fifth grade class, the administrator is presented with another drop down box listing all classes that are one grade and one year ahead of the selected class. The second class is then selected.

Students graduating to the seventh grade will no longer have access to the system, so they are removed. For those who will remain in the system, the first through fifth grade students, several things happen behind the scenes. The selected class to which the students will be promoted cannot already contain students; therefore, this selected class must be cleared of all students, another class must be selected or a new class must be created. If the selected class is empty, the students being promoted to that class will have their 'grade' and 'classid' fields updated to match their new class. And finally, those student's tests and scores from the current year will be removed from the system.

4.5.4 Managing classes

When adding a new class, five fields are requested: 'Class Title', 'Grade Level', 'Year', 'Semester' and 'Teacher'. All fields are required. Note that classes require a teacher, so teachers must be created first. This information is stored in the 'class' table, and a distinguishing auto-number is generated for the 'classid'.

When modifying a class, all classes are listed in a drop down box. Once a class is selected, the modifications can be made. The same fields present when adding a class may be modified, and again all fields are required. The current teacher is listed on-screen. If this teacher has been deleted, a message will indicate such in place of the teacher's name, and a new teacher must be selected. The 'classid' cannot be changed.

When deleting classes, a "Search Type" and "Search Term" are requested. Search types consist of the following: "Year", "Grade", "Name" and "Faculty User Login"; one of these must be selected or the search will not be processed. The software matches the search term to any phrase contained within the search type and lists those matches in a drop down box. All students must be removed from a class before it can be deleted, and once this is done, the class is removed from the 'class' table and all tests assigned for the class are deleted from the 'assigned_test' table.

4.6 Tests and practice tests

Tests are created or generated, assigned and deleted by teachers. Students take the assigned tests and can also generate their own tests for practice outside of class.

4.6.1 Creating tests

This feature consists of two parts: creating the test and creating the questions on the test. Creating the test involves the following. Five fields are required: 'Test Name', 'Grade', 'Description', 'Question Style' and 'Time Limit'. 'Question Style' allows teachers to choose whether questions will appear in a vertical or horizontal fashion, as shown in Figure 6, respectively.

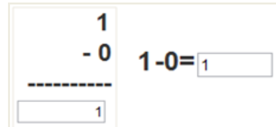


Figure 6: Question Style

Next, the teacher can manually create questions for the test. There are twenty fields for the teacher to create up to twenty questions. A sample field is presented in Figure 7.

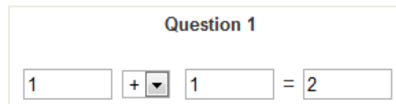


Figure 7: Question Field

Any integer value can be entered in the three text fields. The drop down box contains the four operators: addition (+), subtraction (-), multiplication (X) and division (÷). The answer field is not checked for accuracy; therefore, it is the teacher's responsibility to ensure that correct information is entered. The test is stored in the 'test' table and the questions are stored in the 'question' table.

4.6.2 Generating tests and practice tests

These two features are essentially the same, though when a teacher generates a test, he or she must also name the test, assign it to a grade and describe it. The remaining fields apply to both teachers and students: 'Type of Questions', 'Question Style', 'Time Limit', 'Max Addition and Subtraction Value', 'Max Multiplication Value', 'Multiplication Fact Restriction' and 'Max Division Value'.

The 'Type of Questions' can be addition, subtraction, multiplication, division or any combination of those four. The four max values must be greater than zero, though there is no upper limit. The 'Multiplication Fact Restriction' requires more detail. This feature allows the user to generate tests where the first value in a multiplication problem is limited to the value selected from the 'Multiplication Fact Restriction' field. The options are: zero through ten and no restriction.

Once the information is entered and submitted, the software randomly generates problems within the chosen parameters. Addition and multiplication problems are easily generated. Their only restriction is that the operands cannot be greater than the max specified value (for addition problems, this also applies to the answer); multiplication problems must also follow the specified fact restriction (if any). Subtraction and division are a bit more difficult. Subtraction problems can't yield a negative answer, and each operand cannot be greater than the max specified value. Division problems must yield an integer answer, and the divisor can't be zero. Again, the max specified value must be followed.

4.6.3 Assigning tests and taking them

To assign a test, the teacher selects a grade. All tests created for this grade are displayed in a drop down box; a drop down box listing all of the teacher's classes is also displayed. This allows teachers to assign tests created for varying grades to varying classes, i.e. harder tests can be assigned to lower level classes and vice versa. If that specific class has already been assigned that particular test, the teacher will be given the option to reassign that test. If not, the assignment will complete. In either case, the 'assigned_test' table will be appropriately modified, with 'taken' being given the value "No" for each 'studentid' in 'classid' which has been assigned 'testid'.

When a student wishes to take an assigned test, he or she selects one from a drop down box. When a test is selected, the test screen will be generated according to the student's preferences and test specifications, and testing will commence. An example test screen is presented in Figure 8.

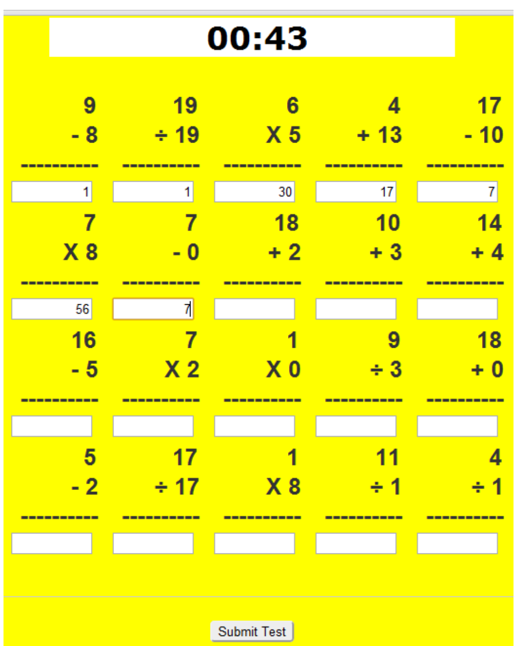


Figure 8: Example Test Screen

All answered questions will be stored in the 'complete_test' table. The student's score and time taken on the test will be stored in the 'score' table, and his or her score and/or time goals will be appropriately updated to reflect his or her performance on the test. Figure 9 shows an example of a performance report, presented after completion of an assigned test:

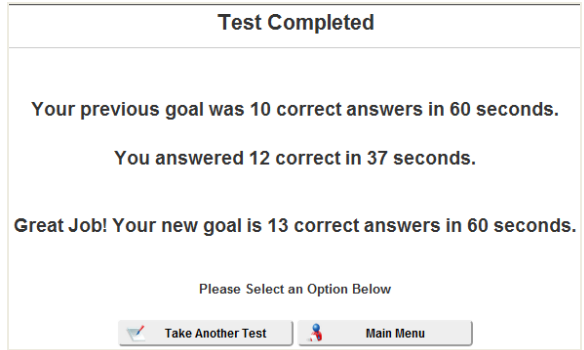


Figure 9: Performance Report

4.6.4 Deleting tests

The teacher selects a grade, and all tests created for that grade are listed in a drop down box. Once the teacher has selected a test, that test and all of its questions are deleted from the 'test' and 'question' tables.

4.7 Viewing test results

All users have access to test results. Administrators and teachers can view test results by class or by test. Students have access to their own test results, though no results are stored for practice tests. Test results can be viewed by two different methods: textual and visual. The information is retrieved from the 'score' table.

4.7.1 View test results by class

This option is available for both administrators and teachers, though the scope of results available to each user type differs. Since administrators are not assigned a class, they are given access to all results for all classes. Teachers are assigned to a class (or classes); therefore, they only have access to the results for their assigned classes. Both must follow a similar process to access results.

First, administrators and teachers must choose between textual or visual results. No matter which option is chosen, the user then selects a class for which he or she wishes to view results. Class results are displayed based on which type of display and which class the user chose.

4.7.2 View test results by test

Administrators and teachers also have access to this function for viewing test results. First, they select a grade, first through sixth. All tests created for the selected grade are listed in a drop down box. Currently, the results for this option are only displayed using the visual method.

4.7.3 View test results for a student

Students have access to this option. It works much the same way as "view test results by class" works for administrators and teachers, with the exception that students do not have the option of selecting a class. They only see results for all assigned tests they have taken.

4.7.4 Textual results

Textual results displays score results as pictured in Figure 10.

Date: 2011-06-09 || Test Name: New Test || Goal: 10 || Number Correct: 12 || Number Incorrect: 8 || Percentage of Goal: 120%

Figure 10: Textual Results

Percentage of goal may not be explicitly clear. It is calculated by equation (1).

$$\text{percentage of goal} = \frac{\text{Number Correct}}{\text{Score Goal}} \times 100 \qquad (1)$$

4.7.5 Visual results

Visual results displays score results as pictured in Figure 11.

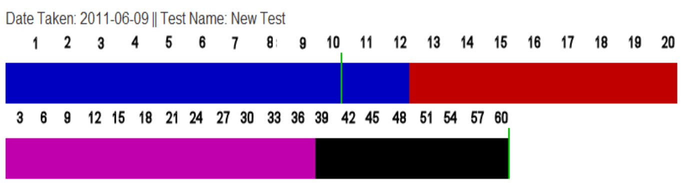


Figure 11: Visual Results

The top visual bar represents the student's score performance for that test. The section of the bar from the left indicates the number of questions the student answered correctly, and the remaining part indicates the number of questions the student answered incorrectly. The thin line extending above the bar (at 10) indicates the student's score goal going into that test.

The bottom visual bar represents the student's time performance for that test. The section of the bar from the left indicates the time taken on that test, and the remaining part indicates the time remaining for that test. The thin line extending above the bar (at 60) indicates the student's time goal going into that test.

5 Current status and future work

MathFax is currently undergoing final modifications and testing. A beta version of the software has been installed at St. Anthony's Catholic School, where students, teachers and administrators at the school will be testing the software for bugs and ease of use. We plan for the software to be fully functional, installed and ready for use before the 2011-12 school year. As is our policy with all of our service-learning projects, after this final testing phase is complete, we will make the software available to any school that wishes to use it.

6 References

[1] Schwartz, Donald R., "Incorporating Service Learning into Capstone/Software Engineering Group Projects", *Proceedings of the 2010 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-10)*, Las Vegas, July 2010.

[2] Schwartz, Donald R., "The Benefits (and Drawbacks!) of "Real World" Software Engineering Semester Projects: A Case Study", *Proceedings of the 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-09)*, Las Vegas, July 2009.

[3] Schwartz, Donald R., "CLOWNS – A Software Engineering Semester Project Currently In-Use in Kindergarten Classes", *Proceedings of the 2008 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-08)*, Las Vegas, July 2008.

[4] Schwartz, Donald R., "Service Learning and Software Engineering: Managing 'Real World' Group Projects", *Proceedings of the 2007 International Conference on Software Engineering and Data Engineering (SEDE-07)*, Las Vegas, July 2007.

[5] Kaye, Cathryn Berger. The Complete Guide to Service-Learning. Free Spirit Publishing. Minneapolis, MN. 2004.

[6] Underwood, Charles, Mara Welsh, Mary Gauvain and Sharon Duffy. "Learning at the Edges: Challenges to the Sustainability of Service-Learning in Higher Education", UC Links. http://www.uclinks.org/reference/research/underetal_01.html.

[7] Schwartz, Donald R., "Managing and Assessing Software Engineering Group Projects", *Proceedings of the 2005 American Society for Engineering Education Annual Conference and Exposition*, June 2005.

[8] Romain Bourdon. "WampServer". http://www.wampserver.com/en/, Last Viewed: May 2011.

Web 2.0 + FOSS: Building Undergraduate Project Management System on Cloud

Cedrik Brown, Hong Jiang, Bethel Tarekegne and Kory Griggs
School of STEM, Benedict College, Columbia, SC, USA

Abstract— *This paper describes a methodology to build undergraduate educational/research project management system on Cloud by combining the technology of Web 2.0 and FOSS (Free and Open Source Software). Specifically, by adopting the free technologies such as Google Sites, Google Docs, Google Talk, Facebook, Twitter, RSS, widgets and OSS (Open Source Software) phpBB and WordPress, we show how these technologies can be embedded to provide appropriate web services to satisfy the needs of an educational or research project management. Two sample websites are introduced to show how these embedded web services are used to provide the actual project management: (1)one is designed for a summer research program using Google site as a base; (2)another is designed for a service-learning project to collect free or low cost technology and serve the community. The features of the major applicable web platforms are also compared and discussed for better application upon individual needs.*

Keywords: Web 2.0, Free and Open Source Software, Project Management, Web service

1. Introduction

FOSS (Free and Open Source Software), as defined in [1], [2], is a very valuable resource that is often overlooked. It has become a major movement in the modern software industry [3], [4]. However, in the education area, other than numerous endeavors [5], [6], [7] to study how to get the students involved in FOSS projects, most educators did not become aware of this valuable low cost resource.

This paper describes a methodology to build undergraduate educational / research project management system on Cloud by combining the technology of Web 2.0[8] and FOSS (Free and Open Source Software). It provides a free/low cost solution for undergraduate educational / research program to monitor and manage their projects. It also shows an example of an undergraduate research project on FOSS.

Specifically, through Web 2.0, we allow users to collaborate with each other in various dimensions, including blogs, forums, social networking sites, etc. Through FOSS, we take the benefits of the freedom of using software in free / low cost and capability of accessing the possible source code. This undergraduate research project adopts free technologies such as Google Sites / Docs / Chat, Facebook, Twitter, RSS, widgets and OSS (Open Source Software) phpBB and WordPress, etc. We show how these technologies can be embedded to provide appropriate web services to satisfy the needs of an educational or undergraduate research project management. Two sample websites are given to show how these embedded web services are used to provide actual project management: (1) The first website is designed to manage a summer research program - Advanced Computing 2010 and 2011 using Google site as a base; (2) The second website is designed to collect free or low cost technology in education based on FOSS and serve the community through online tutorial. The features of the major applicable web platforms are also compared and discussed for better application upon individual needs.

2. Project Management System on Cloud

2.1 Overview

An overview of the undergraduate educational / research project management system is described in Figure 1. It shows the general approaches to combine the technology of Web 2.0 and FOSS.

Fig. 1: Project Management System Overview

2.2 Key Components with Embedded Services

For a project management system, we will need to have the following key components supported by the related embedded web services:

Project Planning: We need to define the project goals, scope and determine the appropriate methods for completing the project. This research adopts the free website platforms such as Google site and WordPress to do the project planning by describing the central idea, structuring the objectives and drafting related tasks, etc. A sample website is in Figure 2.

Organizing / Managing Resources: To facilitate the organization and management of the related project resources, we use RSS, plugins, gadgets or widgets to centralize the resources on the provided website platforms. For example, we can embed resources from different blogs and news by using RSS, and embed desirable services by using plugins / gadgets / widgets, such as calendars, announcements, file cabinet, videos, slideshows, presentations, etc.

Progress Tracking: To monitor and track the project progress, we use blog updates to record the project progress; spreadsheet with chart in google docs to show the progress; and various collaboration tools such as online chat / "contact me" function to facilitate collaborations in a project. We also use free online service Dropbox and shared google docs to enhance the collaboration.

Evaluation: It is very important to collect related data to evaluate a project. This research provide multiple ways to evaluate the project, such as, the embedded online survey form based on the form function in google docs, blog rating and feedback system provided by WordPress, etc.

Security: Considering different levels of security, Google docs and sites provide the sharing functions to (1) keep it private; (2) share with limited collaborators with viewable only or editable authority, or (3) make it public.

As page limits, some embedded web services are marked in Figure 2 (1).

(1)*Summer Research Project Website*

(2)*A Service−Learning Project Forum*

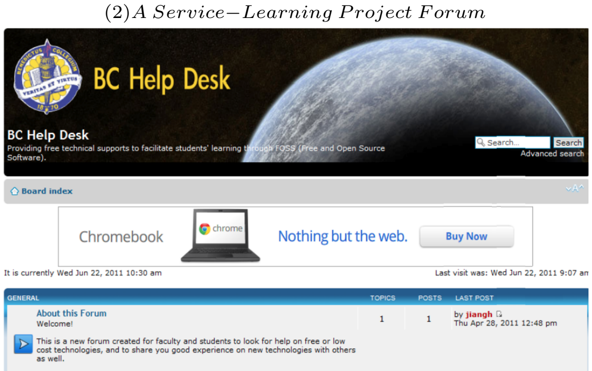


Fig. 2: Resulted Sample Websites

3. Resulted Sample Websites

There are 2 resulted websites as in Figure 2: (1) is designed to manage a summer research program - Advanced Computing 2010 and 2011 using Google Sites; (2) is designed for a service-learning project to collect free or low cost technology and serve the community, other than this forum using phpBB, we also adopt WordPress to centralize different resources and provide multiple functions for collaboration.

4. Website Platforms Comparison

There are three major website platforms that have been considered in our research. The major features are compared as in the following Table 1. According to individual needs, a user can choose the desired website platform and embedded services for a project. In our resulted sample websites, we prefer WordPress and Google Sites because of the flexibility to embed different services.

Table 1: Comparable Website Platforms

Usage	WordPress	Google Sites	phpBB
Easy to use / setup	√	√	√
No html coding required	√	√	√
Themes / templates	√	√	√
Website URL	Short	Long	Short
Open Source	√	×	√
Extended functionality with plugins / gadgets / widgets or coding	Best	Good	Limited
Connection with twitter / facebook	Automatically	Manually, Limited	×
Sharing files	×	√	×
Rating / comments / feedback	√	Sign in required	Sign in required
Capability to install on user's server	√	×	√
The best feature	Blog	Multifunctions	Forum
Cost	Free	Free	Free

5. Conclusions

This undergraduate research provides a methodology to build project management system on Cloud by combining the technology of Web 2.0 and FOSS. It provides a free/low cost solution for undergraduate educational / research program to monitor and manage their programs.

References

[1] GNU.org, "The free software definition," http://www.gnu.org/philosophy/free-sw.html, November, 2010.
[2] Opensource.org, "The open source definition," Open Source Initiative. http://opensource.org/docs/osd, 2011.
[3] T. Claburn, "Study finds open source benefits business," InformationWeek. CMP Media LLC. http://www.webcitation.org/5TchF5fkl, 2007.
[4] Flossproject.org, "Free/libre and open source software: Survey and study," International Institute of Infonomics. http://flossproject.org/report/index.htm, 2002.
[5] TOSS09, "Teaching open source summit," Teaching Open Source. http://teachingopensource.org/index.php/TOSS09, 2009.
[6] A. Tridgell and B. Edwards, "Comp8440: Free and open source software development," The Australian National University. http://cs.anu.edu.au/student/comp8440/, 2011.
[7] W. Shroder, "Open source software practice," Rensselaer Center for Open Source Software. http://public.kitware.com/OpenSourceSoftwarePractice/, 2011.
[8] T. O'Reilly, "What is web 2.0: Design patterns and business models for the next generation of software," O'Reilly, http://oreilly.com/web2/archive/what-is-web-20.html, 2005.

Learning Objects and Ontologies to Perform Educational Data Mining

F. Castro, M.A. Alonso
Research Center on Systems and Information Technologies, Autonomous University of Hidalgo's State, Pachuca, Hidalgo, Mexico
fcastro@lsi.upc.edu, marial@uaeh.edu.mx

Abstract - *E-learning systems have established as a strong alternative to traditional distance education. One of the most valuable, but unfortunately, less used in online educational courses is the learning objects (LO) technology. Internet allows the gathering of plenty of information on students' online behavior; however, this information is in raw format, making difficult the knowledge extraction. Moreover, information about the usability of components in the online course, LO, is rarely obtained, and therefore, hardly used in the data mining (DM) process. The knowledge extracted from this information can be used to define personalization strategies tailored to the students' needs and requirements. In this brief study we introduce a platform to perform educational DM process based on gathered information from both, the students' navigational activities in the e-learning system and the information collected from the LO usability. Moreover, the gathered data are structured, organized and formalized by means of an educational ontology.*

Keywords: E-learning, Data Mining, Learning Objects, Ontology.

1 Introduction

Within a decade, the Internet has become a pervasive medium that has changed completely the way information and knowledge are transmitted and shared all over the world. The education community has not limited itself to the role of passive actor in this unfolding story, but it has been at the forefront of most of the changes. Indeed, the Internet and the advance of telecommunication technologies allow us to share and manipulate information in nearly real time. This reality is determining the next generation of distance education tools. Distance education arose from traditional education in order to cover the necessities of remote students and/or help the teaching-learning process, reinforcing or replacing traditional education.

This is e-learning, a new context for education where large amounts of information describing the continuum of the teaching-learning interactions are continuously generated and ubiquitously available. This could be seen as an approval: a lot of information readily available just a click away. But it could equally be seen as an exponentially growing nightmare, in which unstructured information strangles the educational system without providing any articulate knowledge to its actors.

Data Mining was born to tackle problems like this. As a field of research, it is almost contemporary to e-learning. Not just of its intrinsic complexity, but because it has most of its roots in the ever-shifting world of business. At its most detailed, it can be understood not just as a collection of data analysis methods, but as a data analysis process that encompasses anything from data understanding, pre-processing and modelling to process evaluation and implementation [1]. It is nevertheless usual to pay attention to the Data Mining methods themselves.

Therefore, Data Mining can be used to extract knowledge from e-learning systems through the analysis of the information available in the form of data generated by their users. In this case, the main objective becomes finding the patterns of system usage by teachers and students and, perhaps most importantly, discovering the students' learning behavior patterns. This process is known as educational data mining.

Several research projects have dealt with the integration of data mining methods focusing on e-learning systems improvement. For a deeper inside into these projects the authors recommend [2, 3, 4], where a widespread and deep analysis on different learning platforms is performed, including LON-CAPA [5], AHA! [6], ALFANET [7], etc. Commonly, the existing platforms perform students' classification (using supervised neural networks, decision trees, fuzzy methods, association rules, etc.), and/or students' clustering (using Kohonen's self-organizing maps, EM, etc.).

Most of this research uses data mining techniques to analyze the available data. In turn, we can distinguish diverse Soft Computing-based approaches to e-learning process analysis, i.e. methods to classify students' based on their usage patterns on a web-based course [8, 9, 10], methods oriented towards system personalization [11]. For instance, a neural network model is proposed in [12] to recommend an adequate navigation strategy for the user. A methodology to improve the performance of developed courses through adaptation, using Evolutionary algorithms, is presented in [13]. And, finally, methods that allow automatic detection of atypical students' behavior such as the Bayesian predictive distribution model to detect irregular learning proposed in [14], and the Generative Topographic Mapping model to detect atypical behavior on the grouping structure of the users of a real virtual campus, presented in [15].

However, just a short number of papers have devoted to include ontologies as a support to perform educational data mining processes, for instance, in [16], a framework for personalized e-learning based on aggregate usage profiles and a domain ontology were presented and a combination of Semantic Web and Web mining methods was used. The Apriori algorithm for Association Rules was applied to capture relationships among URL references based on the navigational patterns of students.

An ontology-based tool, within a Web Semantics framework, was implemented in [17] with the goal of helping e-learning users to find and organize distributed courseware resources. An element of this tool was the implementation of the Bisection K-Means algorithm, used for the grouping of similar learning materials. An approach to automate the classification process of Web learning resources was developed in [18]. The model organizes and labels learning resources according to a concept hierarchy extracted from the extended ontology of the ACM Computing Curricula 2001 for Computer Science.

Despite the existence of some research concerned with the mining of data generated by the use of e-learning systems, there is still a lack of standard methods and techniques to address some open problems in distance education. This is the case, for instance, of methods that could detect similarities in learning behaviors and group (cluster) students according to them. This strategy could be used, for instance, to provide teachers with an adaptive guidance tool to prevent student failure.

It is very difficult and time consuming for teachers to thoroughly track and assess all the activities performed by all students. Moreover, it is hard to evaluate the structure of the course content and its effectiveness on the learning process as it develops. Therefore, it would be helpful to obtain objective feedback from learners in order to track the learning process and assess the online course structure effectiveness [19].

The possibility of tracking user behavior in such environments creates new possibilities for both web-based system architects and designers, but also for the pedagogical and instructional designers who create and organize the learning contents. One of the most interesting possibilities is the personalization of the e-learning process. Personalization arises from the knowledge extracted from the navigational behavior of the e-learning virtual environment users, mostly students in this particular scenario.

To deal with the majority of the problems addressed before, in this brief study we propose the integration of learning objects (LO), including an educational ontology, as a support, in the educational data mining process. The LO can, both, generate information about their usability using the metadata and, by means an ontology, provide a formalization of the entities from the e-learning domains. Learning objects are any digital resource that can be reused for learning or training, and constitute a valuable resource for e-learning course development. Moreover, the motivation for developing an ontology for educational data mining is many-fold. On the one hand, the educational data mining area is devoting many research efforts and rapidly increasing; however, one of the most challenging problems deals with developing a general framework for educational data mining. On the other hand, often, the data extracted from the educational system are raw and unstructured making difficult the knowledge extraction and the decision support stage. Moreover, the students could increase their course performance if the teachers properly provide them with a suitable and actionable feedback; this activity can be achieved by means an appropriate ontology. An educational ontology can be used for formalizing and describing educational scenarios.

The remaining of the paper is organized as follows: section 2 presents a description of our educational data mining proposal. Finally, section 3 wraps up the paper with some preliminary conclusions.

2 Proposed Educational Data Mining Approach

As has stated before, one of the first and more difficult steps in the data mining process is the pre-processing, cleaning and integration data, due to the fact that often the gathered data are in a raw and unstructured format, even, including mixing data.

In the proposed approach, the educational data mining process would benefit from the formalization of data and educational entities provided by an educational ontology. The proposed ontology can be used to discover relationships between the learning resources metadata and should follows the best practices in ontology engineering, for instance, it must be a deep/heavy-weight ontology avoiding multiple inheritance of classes, using a predefined set of relationships and using a top level ontology. Furthermore, the integration of learning objects (LO), and an educational ontology can, both, generate information about the usability of the LO and provide a formalization for the entities from the e-learning activities. Learning objects provide valuable support to deal with the standardization, interoperability, cooperation, transportation and reusability of educational content. The proposed educational ontology would helps and standardizes the data extraction and preparation in the educational data mining process and makes easier the knowledge extraction. Furthermore, the integration of LO and educational ontology will provide valuable support in the students' feedback activity, to help them to cover their academicals needs and requirements.

For the sake of space availability in Figure 1 we summarize the interaction between the actors and the virtual campus as well as the functionalities offered by the proposed platform. The main goal of the platform is to alleviate the virtual tutors' workload and to provide an effective and valuable feedback to students. To deal with these objectives the platform offers tools to discover relevant learning behavior patterns from students' interaction with the educational materials. The knowledge obtained can be used by teachers to design courses in a more effective way and to timely detect students with learning difficulties. The extracted knowledge can also be helpful for the students to know their own learning performance and therefore use more efficiently the educational resources.

The platform presented in this paper includes as part of its skills the identification of students' learning behavior models that allow both, students and teachers, to know the future performance of the student based on their current learning behavior and the course information available at that moment. This functionality allows teachers to give actionable feedback to those students that need it, such as the failing students. The platform also provides functionalities to forecast students' learning performance and to determine the most relevant features involved in the evaluation process, and their relative weight. Finally, it allows extracting logical rules that describe the students' learning behavior patterns, which are easily understandable by experts in educational domains. All the platform functionalities could be performed when the teacher or student considers relevant doing with the main characteristic that in each execution will be used the information available at the moment, i.e. using temporal models that, of course, give better performance prediction results when it has wealthy information that often happens at the last phases of the course. Using the information obtained by the toolbox, the teachers could prevent possible course failing to the students susceptible to obtain worse performance results in order to accomplish their learning requirements.

3 Preliminary Conclusions

The tracking of user navigational behavior in virtual campus e-learning environments makes the web mining of the resulting data bases possible. This opens new possibilities for the pedagogical and instructional designers who create and organize the learning contents; amongst the most interesting ones, the personalization of the e-learning process.

One of the most difficult and time-consuming activities for teachers in distance education courses is the evaluation process, due to the fact that, in this type of courses, the review process is better accomplished through collaborative resources such as e-mail, discussion forums, chats, etc. As a result, this evaluation usually has to be carried out according to a large number of parameters, whose influence in the final mark is not always well defined and/or understood. Therefore,

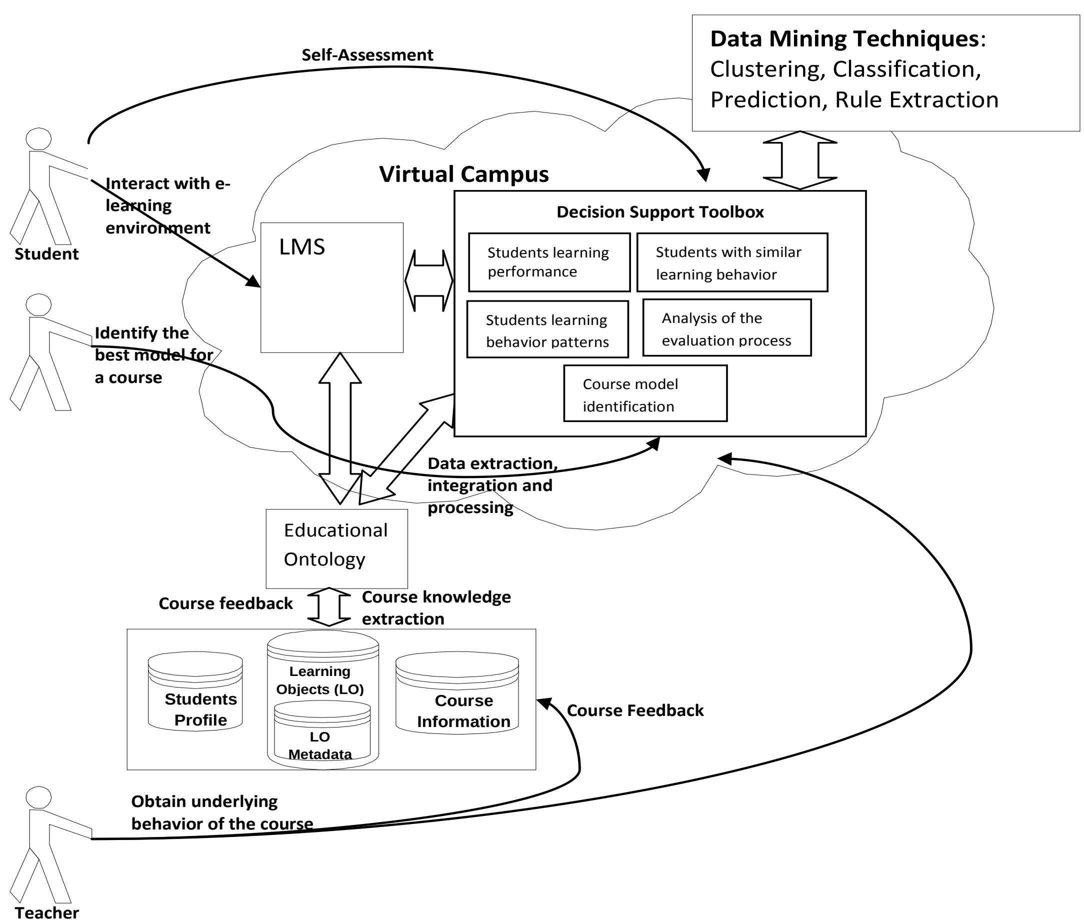


Figure 1. Proposed platform functionality.

it would be helpful to discover features that are highly relevant for the evaluation of groups (clusters) of students. In this way, it would be possible for teachers to provide feedback to students, on their learning activity, online and in real time in a more effective way than if we tried to do it individually. This option, using clustering methods, could reduce teachers' workload. In this sense the grouping of students with similar learning behavior could significantly help in the e-learning personalization, by the fact that the teacher can send grouped students' feedback instead to perform that in an individualized way, alleviating their workload, and even, the learning groups' behavior can be used to create greatly cohesive, organize and interactive students group based on their performance and/or learning behavior.

In order to obtain concluding remarks, nowadays, we are performing a set of deeply and detailed experiments with the aim to identify qualitatively and quantitatively the course improvements, accomplished using the resulting knowledge provided for the proposed platform. Partial results, obtained by the authors, have indicated that the teachers workload have been reduced considerably, optimizing their course activities, and providing a better quality of the students' feedback; accordingly the students learning and course performance have significantly increased. As a future trend of the presented platform, we would include a Computer Adaptive Tests (CAT) methodology in order to improve the e-learning evaluation process. We think that this could significantly help teachers in the evaluation procedure due to the fact that this methodology could provide valuable teacher support in reducing the e-learning teacher workload. Preliminary results of the authors are very encouraging in this educational topic.

Acknowledgements

The authors gratefully acknowledge funding from Mexican Secretary of Public Education through the project FOMIX-HGO-2009-C01-131024.

4 References

[1] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRIPS-DM 1.0 Step by Step Data Mining Guide. CRISP-DM Consortium (2000).

[2] Castro, F., Vellido, A., Nebot, A. & Mugica, F. (2007). Applying Data Mining Techniques to e-Learning Problems. In L.C. Jain, R.A. Tedman & D.K. Tedman, Evolution of Teaching and Learning Paradigms in Intelligent Environment. Studies in Computational Intelligence, 62 (pp. 183-221). Germany: Springer-Verlag.

[3] Croock, M., Mofers, F., Van Veen, M., Van Rosmalen, P., Brouns, F., Boticario, J., Barrera, C., Santos, O., Ayala, A., Gaudioso, E., Hernández, F., Arana, C., Trueba, I. (2002). State-of-the-Art. ALFanet/IST-2001-33288 Deliverable D12. Open Universiteit Nederland. URL: http://learningnetworks.org/

[4] Van Rosmalen, P., Brouns, F., Tattersall, C., Vogten, H., van Bruggen, J., Sloep, P., Koper, R. (2005). Towards an Open Framework for Adaptive Agent-Supported e-Learning. International Journal Continuing Engineering Education and Lifelong Learning 15(3–6) 261-275.

[5] Minaei-Bidgoli, B., Tan, P.N., Punch, W.F. (2004). Mining Interesting Contrast Rules for a Web-based Educational System. In Proceedings of The 2004 International Conference on Machine Learning and Applications, ICMLA'04. Louisville, KY.

[6] Romero, C., Ventura, S., De Bra, P., De Castro, C.(2003). Discovering Prediction Rules in AHA! Courses. In Proceedings of the User Modelling Conference. Johnstown, Pennsylvania (pp. 35-44).

[7] Van der Klink, M., Boon, J., Rusman, E., Rodrigo, M., Fuentes, C., Arana, C., Barrera, C., Hoke, I., Franco, M. (2002). Initial Market Study, ALFanet/IST-2001-33288 Deliverable D72. Open Universiteit Nederland. URL: http://learningnetworks.org/downloads/alfanet-d72-initialmarket -studies.pdf.

[8] Minaei-Bidgoli, B. and Punch, W.F. (2003). Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System: Genetic and Evolutionary Computation Conference.

[9] Mullier, D. (2003). A Tutorial Supervisor for Automatic Assessment in Educational Systems: International Journal on E-Learning.

[10] Stathacopoulou, G.D. and Grigoriadou, M. (1999). Neural Network-Based Fuzzy Modeling of the Student in Intelligent Tutoring Systems: International Joint Conference on Neural Networks, Washington.

[11] Brusilovsky, P. (2001). Adaptive hypermedia: User Modeling and User Adapted Interaction, Ten Year Anniversary Issue, 11 (1/2), 87-110.

[12] Mizue, K. and Toshio, O. (2001). N3: Neural Network Navigation Support–Knowledge-Navigation in Hyperspace: The Sub-symbolic Approach: Journal of Educational Multimedia and Hypermedia 10(1), 85-103.

[13] Romero, C., Ventura, S. and De Bra, P. (2004). Knowledge Discovery with Genetic Programming for Providing Feedback to Courseware: User Modeling and User-Adapted Interaction, 14(5), 425-464.

[14] Ueno, M. (2003). LMS with irregular learning processes detection system: World Conference on E-Learning in Corp., Govt., Health, & Higher Ed., 2003(1), 2486-2493.

[15] Castro, F., Vellido, A., Nebot, A. and Minguillón, J. (2005). Detecting Atypical Student Behaviour on an e-learning System: VI Congreso Nacional de Informática Educativa, Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE'2005). Granada, España, September 14-16.

[16] Markellou, P., Mousourouli, I., Spiros, S., Tsakalidis, A.: Using Semantic Web Mining Technologies for Personalized e-Learning Experiences. In: Uskov, V. (ed.): The Fourth IASTED Conference on Web-based Education, WBE-2005. Grindelwald, Switzerland (2005)

[17] Tane, J., Schmitz, C., Stumme, G.: Semantic Resource Management for the Web: An e- learning Application. In: Fieldman, S., Uretsky, M. (eds.): The 13th World Wide Web Conference 2004, WWW2004. ACM, New York (2004) 1–10

[18] Singh, S.P.: Hierarchical Classification of Learning Resources through Supervised Learning. In: World Conference on e-Learning in Corp., Govt., Health., & Higher Education (2004) 178–183

[19] Zaïane, O.R. and Luo, J. (2001). "Towards evaluating learners' behavior in a Web-based distance learning environment", in Proc. of IEEE International Conference on Advanced Learning Technologies (ICALT'01), Madison, WI, August 6-8, pp 357-360.

@rabLearn: Towards Intelligent Answer Analysis and Feedback in ICALL

Mourad Mars[1], Georges Antoniadis[1], Mounir Zrigui[2]

[1] *LIDILEM Laboratory Grenoble - France*
Mourad.mars@e.u-grenoble3.fr
Georges.antoniadis@u-grenoble3.fr
[2]*UTIC Laboratory Monastir - Tunisia*
Mounir.zrigui@fsm.rnu.tn

***Abstract*— This paper describes the development of an intelligent computer-assisted language learning (ICALL: Intelligent Computer Aided Language Learning) system for learning Arabic, especially, the advanced method of feedback and the approach for analysing learner's answer. The corrective feedback is undoubtedly the attribute generally associated with the ICALL and the interactive language learning environment, it forms the essential component to evaluate the quality of a teachware. As it is the case for other concepts, several traditions converge to lead to the concept of feedback in ICALL: we talk about feedback in the area of the technology, the area of information and communication and other fields. After a brief presentation of the general denomination of this word, we show what the theories of learning languages say on the feedback and the correction of errors produced by learners. We will study aspects of feedback specific to the field of ICALL by describing our language learning environment called @rabLearn which is based on the use of NLP (Natural Language Processing) tools such as our Morphological analyser [7, 8] and we seek to know in what ways analysis of learner's answers by the computer improve the quality of the feedback.**

***Keywords* — ICALL, Natural Language Processing, Arabic Language Learning, Feedback, CEHL.**

I. Introduction

The introduction of the TIC (Technology of Information and Communication) into languages learning platforms seems subjected to the need for for close monitoring of technological advances. Although a transformation and an improvement of the output always levelled, the meticulous observation of the multiple devices and software which we carried revealed faults, which only the use of the NLP is able to correct it [9, 10, 12]. In this context, our approach can be expressed by: how could systems of ICALL and in a general context the computer environment of human learning (CEHL.), benefit from procedures, solutions and tools of natural language processing? [16, 19]. Some platforms dedicated to languages learning, like @rabLearn for Arabic language (under development at LIDILEM and UTIC Laboratories), provides answers to this question. @rabLearn allows the creation of educational activities and it facilitate the conception and enhance transparency to teachers of Arabic language to conceive several activities using NLP tools; this process is made with simple and very ergonomic graphical interfaces adapted to all teachers even those without any knowledge. This approach is intended for qualified teachers of language or teachers having only little or no knowledge in NLP or in computer sciences. The technical side is transparent for the user and only the didactic and pedagogic is visible and available.

In other papers, we have explained how we can integrate an NLP tool like our morphological analyser [7, 8, 9] in a platform of learning to generate activities intended for learner

[9]. We have proved that this tool can make platforms more powerful by creating various kinds of activities.

In the field of the interaction teacher-learner and ICALL in general, the feedback holds particular place. Indeed, it is considered as one of the essential factors of language learning. In this paper, we will study the aspects and kinds of feedback specific ICALL, and we will seek to know in what ways the analysis of the of learner's answer by the computer improve the quality of the feedback. [15, 17, 18].

II. Feedback and Errors Correction in ICALL

A. *A. Feedback of the teacher and feedback of the computer: brief comparison*

Feedback aiming at the correction of learner's errors presents many important differences according to whether the learner is lavished by the teacher in classroom or by a language learning environment [13, 14]. A good amount of these differences were presented as advantages of the teaching machine. The following table contrasts these positive factors of the feedback in situation of CALL with the weaknesses of error treatment pratices by teachers in classroom situation. [5,6]

TABLE I

COMPARISON OF THE FEEDBACK IN ICALL AND IN CLASSROOM SITUATION

ICALL Feedback	**Classroom Feedback**
Applies to each incorrect response.	Teacher correct error in one part of the lesson and forgot it in another.
Private	Public
Instantaneous	Instantaneous in oral interaction, delayed when written work is corrected.

B. Conception of the feedback

Many theorists [1, 2, 3, 4] and experts in ICALL drew up lists of "principles" intended to guide the design of the ICALL. We extracted from these lists some instructions those seems most important and was integrated in our ICALL environment called @rabLearn (Environment for learning Arabic as second language).

1. Communicative CALL will not try to judge and evaluate everything the student does.
2. Communicative CALL will avoid telling students they are 'wrong'. If errors are to be processed at all, the program should endeavour to help the student, either by modelling a well-formed response or by giving gentle hints in that direction.
3. Communicative CALL will not try to 'reward' students with congratulatory messages, lights, bells, whistles, or other such nonsense.
4. Communicative CALL will not try to be 'cute'.
5. Communicative CALL will be flexible computers do not have to be any more 'rigid and inhuman' than the people who program them.
6. Feedback should focus on content and meaning.
7. Feedback should not be used for testing.
8. Feedback should be communicative and low-keyed. [...] Informal, easy-going responses help to make the lesson communicative.
9. Feedback should differentiate between students.
10. Feedback should be clear, yet economical.
11. Feedback should be simple. One must anticipate the possible difficulties and prevent student's frustrations.
12. Feedback should use a playful tone when it fits naturally.
13. Feedback should avoid set patterns.
14. Feedback should be open to user-driven changes. [Students] call our attention to alternate answers that we hadn't anticipated.

C. @rabLearn: General architecture

Figure 1 shows the overall architecture of the proposed Arabic ICALL system called @rabLearn. This system consists of the following components: user interface, course material, scripts for generation of several activities based on NLP tools and feedback. The user interface provides the means of communications between the learner or teacher and the Arabic ICALL system. The course material includes educational units, an item (question) bank. Scripts are modules that integrate resources or NLP treatment and which have a general educational goal. For example, automatic generation of any activity is considered as a script, which must be set by the teacher as necessary to achieve a specific educational objective. The creation of scripts demands skills in NLP, in computer science and linguistic. NLP programs used here (Morphological Analyser for Arabic [7, 8]) are purely our personal achievement. Given the technical nature of the scripts, the internal interface of this level is not visible to language teachers. Each script is created and identified by a name that should suggest the pedagogical potential holdings. The feedback component includes an intelligent error analyser that is used to parse ill-formed learner input and to issue feedback to the learner. The design of activities and feedback ideas are task carried out by teachers of languages through a specific interface for each kind of activity.

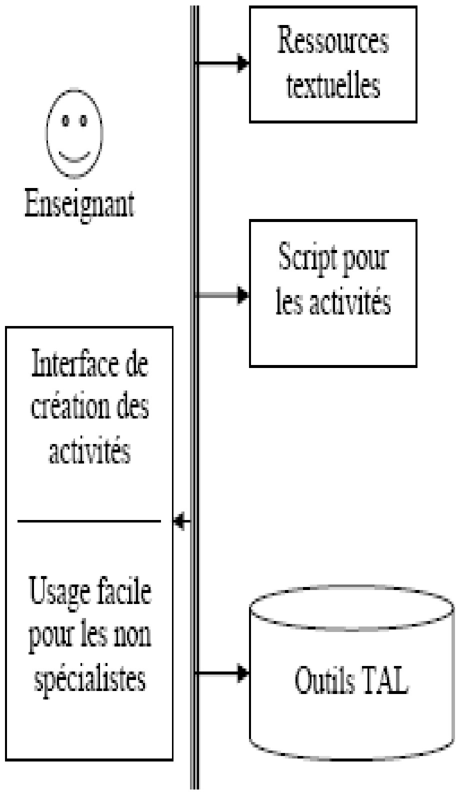


Fig 1 Architecture of our platform for learning Arabic language

@rabLearn platform use NLP tools which appears promising for the linguistics and language teaching [1, 2, 14]. Its main advantage is the simplicity of use; it allows the teacher of the languages, non-NLP specialist [3, 7], to constitute linguistic resources (using simple interfaces) and to parameterize them in order to generate activities intended for learning. To develop any activity using @rabLearn, it is thus not necessary to resort to a complex data processing sequence, or to call upon specialists in data processing or NLP. However the activity is generated, learner can work on and profits from a feedback, this is depends on kind of activities. The next section exposes some kind of activities and feedback.

D. @rabLearn: Generation of Activity with the use of a morphological analyser (NLP tool)

In this section, we expose the steps of generation activities in @rabLearn with the use of an NLP tool.

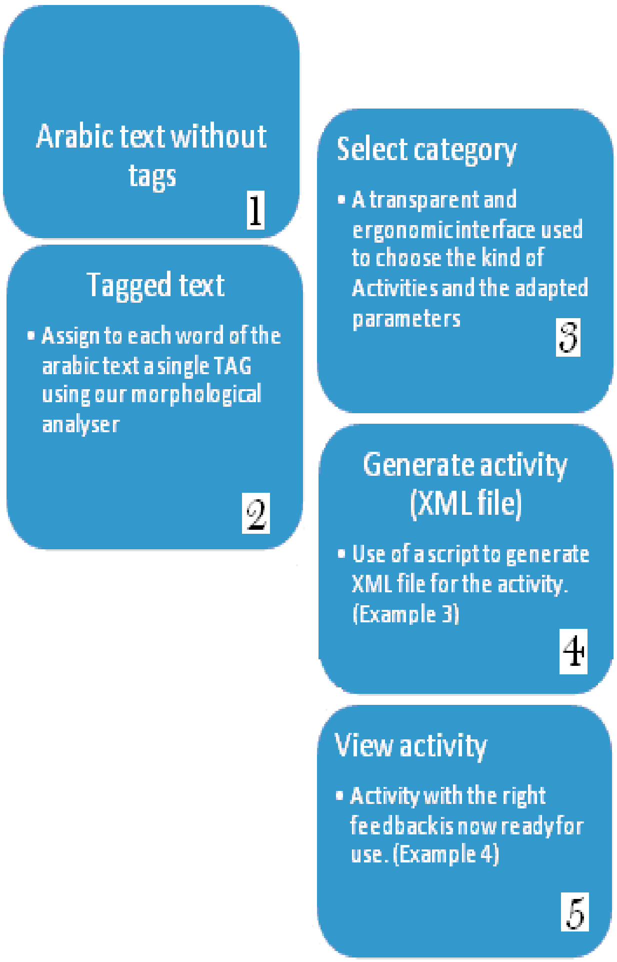


Fig 2 Steps of activities generation.

Example of XML file for text-hole (fill in the blancks) activity:

```
<tok id="t2" base="gwm" ctag="VBP" visibility="non">yqwm</tok>
<tok id="t3" base="AHd" ctag="NOM" visibility="oui">AHd</tok>
<tok id="t4" base="EnSr" ctag="NOM" visibility="oui">EnASr</tok>
```

Fig 3 Activity in XML format

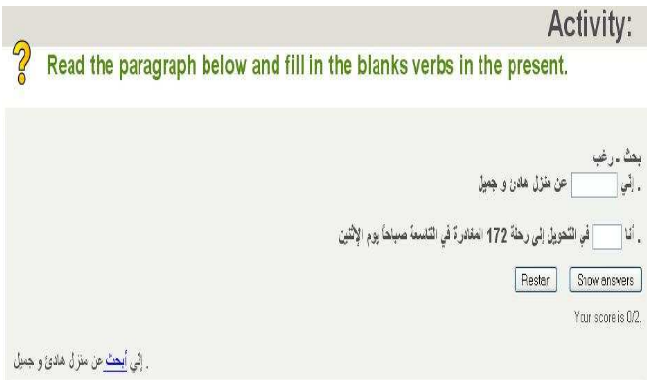


Fig 4 Activity in final format for learner (1)

This activity is now ready for use, learner can start answer the question. An intelligent feedback will start.

Another example of an activity:

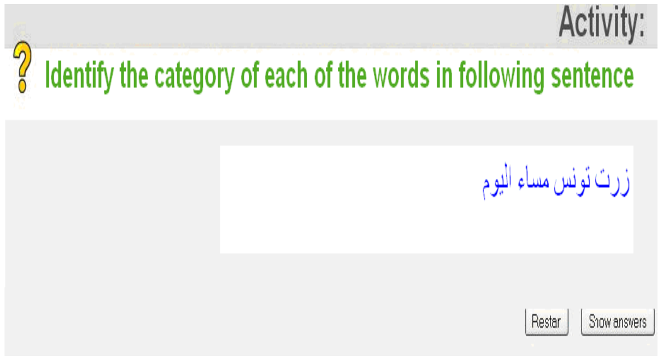


Fig 5 Activity in final format for learner (2)

Identify the category of each of the words in following sentence:

- I Visited Tunis this evening.

The morphological analyser is used to analyse each Arabic word and to recognize its category, it associate to each word of a sentence the right Tag [7, 8] Then, the activity is proposed to learner.

The learner's answer is analysed. A comparison between the corresponding answer will issue the appropriate feedback that describes the source of the error. The possible source of the errors could be: missing words from the respective morphological category.

E. Types of activities and feedback

In order to establish a classification of the principal types of existing activities and feedback in @rabLearn, we propose this table to classify the types of asked questions by the ICALL system, which condition the possible types of answers for learning and - at the third time of the interaction - the feedback itself [14].

TABLE II

Principal types of activities and feedback in @rabLearn

Level	**Question**	**Response**	**Feedback**	**New answer**
1	Text-hole	True or False	Correct answer: Positif feedback Incorrect answer: Negative feedback	Guidance towards a new choice
2	True or False	True or False	Correct answer: Positif feedback Incorrect answer: Negative feedback	Only one possibility

3	QCM	Choice of an answer among other	Correct answer: Positif feedback Incorrect answer: Negative feedback	Many possibility (limited in number)
4	Lacunar exercise	Answer limited to one or more words in preset context	Correct answer: Positif feedback Incorrect answer: Negative feedback	Guidance towards a new choice (in a restricted unit but not limited)
…..				

F. Feedback interaction in @rabLearn activities

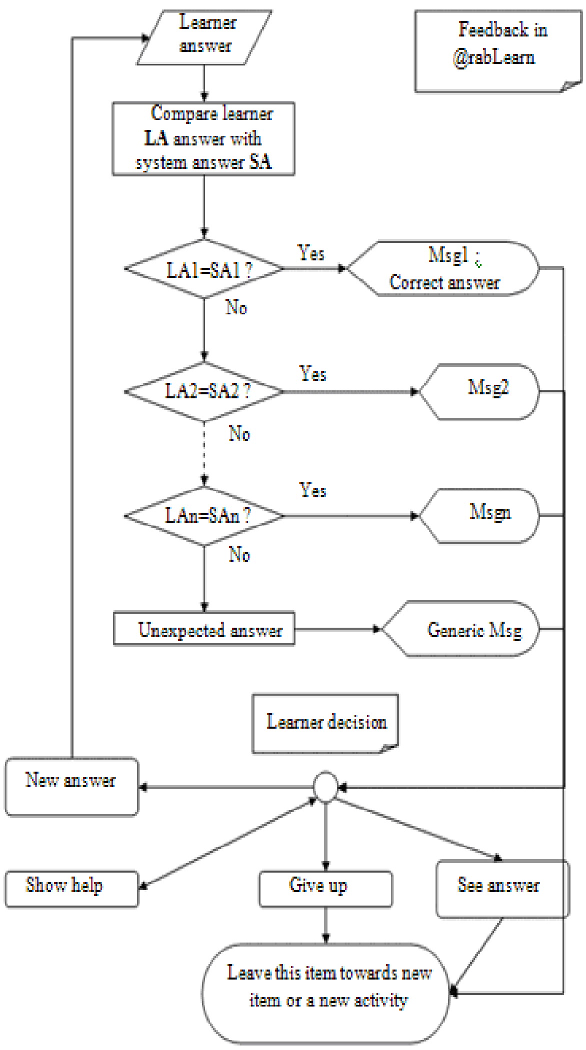


Fig 5 Diagram of Answer analysis and interaction in @rabLearn.

In this section we explain with a diagram the mechanism of answer analysis in @rabLearn, our approach is based on helping learner to find the right answer with the use of an interactive feedback, starting by giving some indicator to guide learner. Our intervention depends on activities and user answer[13]. We called this kind of dialog '*cooperation sequence*'.

G. Quality of feedback

According to Brent [2], the four crucial questions which arise concerning the feedback in ICALL are: which types of feedback are most effective? Which are those which learner prefer? In a language learning environment, in which form learner prefer to receive the feedback? And finally, what learner make with feedback? [11]

To develop @rabLearn, some questions were given to learner in order to take learner answers in consideration and adapt our system to learner preferences.

TABLE IIII

Questionary results.

Method	Number	%
Communicative approach	46	34.1%
Integrative approach	35	25.9%
Situational method	28	20.7%
Functional approaches	14	10.4%
Approaches centred on comprehension	12	8.9%

III. Conclusion

In conclusion, the ordinary NLP tools can be diverted and handled to propose a reflexive approach in languages learning and linguistics. However, in spite of its obvious interest, NLP does not adapt in an immediate way to ICALL systems. The first difficulty is related to the imperfect performances of the NLP tools which must be used with caution in the learning applications. Indeed, if some simple applications (as morphological analysis) give good performances, they are never free from error.

In second part, the relative questions with the design of the feedback in ICALL is treated and integrated in @rabLearn. The diversity of activities, the variety and quality of feedback make @rabLearn a robust environment for Arabic language learning.

We plan to enrich the present system, e.g. make the system available on the Internet to serve remote learners worldwide (especially learners of Arabic as a second language), and extend the grammar coverage to include more advanced grammar levels.

Other important point is: The dimension relates to communication and interaction between the tutor and the learner as shown in table III. Especially important in this interaction is the aspect of a bi-directional interaction: The learner does not only want to have a tutorial feedback but wants to give feedback to the tutor as well. This kind of interaction must be integrated in our next version of @rabLearn.

ACKNOWLEDGMENTS

The morphological analyser used, as an NLP tool, in this project is part of three large research projects, the first one titled “Oreillodule” which is a tool for automatic speech recognition, translation, and synthesis for Arabic language. The others are “MIRTO” & “@rabLearn” which presented in this paper: two Intelligent Computer Assisted Language Learning (ICALL) platforms. The integration of our POS tagger in both ICALL systems will help teachers to generate, automatically, pedagogical activities and enjoy a powerful feedback.

REFERENCES

[1] Allison Piguet, Daniel Peraya. Creating Web-integrated learning environment: An analysis of WebCT Authoring Tools in Respect to Usability. Australia Journal of Educational Technology, Vol. 16, No. 3, pp. 302-314. 2000

[2] Antonija Mitrovic, Brent Martin. Evaluating Effectiveness of Feedback in SQL-Tutor. In the Proceedings of International Workshop on Advanced Learning Technologies, IWALT 2000, pp. 143-144, Palmerston North. 2000

[3] Dominique Hémard & Steve Cushion. Designing a CALL Package for Arabic While Learning the Language Ab Initio, Computer Assisted Language Learning (CALL): An International Journal, Vol. 16(2-3): 259-266, SWETS & ZEITLINER publisher. 2003

[4] Everhard Ditters, Nelleke Oostdijk, and Keith Cameron. Processing Arabic: Computer Assisted Language Learning, Intellect Books, Oxford, England. 1993

[5] Johann Gamper, Judith Knapp. A Review of Intelligent CALL Systems. Computer Assisted Language Learning (CALL): An International Journal, 15(4), 329-342, Belgium: SWETS & ZEITLINGER publisher, 2002.

[6] Merryanna L Swartz, Masoud Yazdani. intelligent tutoring systems for foreign language learning, chapter Introduction. Springer-Verlag, 1992

[7] Mourad Mars, Georges Antoniadis, Mounir Zrigui. Which Algorithm and Approach for Arabic Part Of Speech Tagging. CITII 2010. 27-29 octobre 2010, (Journal Research in Computer Science), Apizaco, Mexique. 2010

[8] Mourad Mars, Georges Antoniadis, Mounir Zrigui. Statistical Part of Speech Tagger for Arabic Language. ICAI'10 - The 2010 International Conference on Artificial Intelligence. 12-15 juillet 2010, Las Vegas, Nevada, USA. 894-899. 2010

[9] Mourad Mars, Georges Antoniadis, Mounir Zrigui, "Nouvelles ressources et nouvelles pratiques pédagogiques avec les outils TAL", TICEMED 08, Journal Information Sciences for Decision Making (Journal ISDM), ISDM32, N°571, Avril 2008.

[10] Mohsen Maraoui, Georges Antoniadis, Mounir Zrigui: CALL System for Arabic Based on Natural Language Processing Tools. IICAI 2009: 2249-2258.

[11] Nielsen H. and Carlsen M. Interactive Arabic Grammar on the Internet: Problems and Solutions, Computer Assisted Language Learning (CALL): An International Journal, Vol. 16(1):95-112, SWETS & ZEITLINER publisher. 2003

[12] Olivier Kraif , Georges Antoniadis, Sandra Echinard, Mathieu Loiseau, Thomas Lebarbé, Claude Ponton . NLP tools for CALL: the simpler, the better. In Proceedings of InSTIL/ICALL2004 Symposium, NLP and Speech Technologies in Advanced Language Learning Systems. 2004

[13] Shaalan F. Kaled.: An Intelligent Computer Assisted Language Learning System for Arabic Learners, Computer Assisted Language Learning: An International Journal, 18(1 & 2), PP. 81-108, Taylor & Francis Group Ltd., February 2005.

[14] Shaalan Khaled F., Habib E. Talhami. Arabic Error Feedback in an Online Arabic Learning System1. Advances in Natural Language Proccesing, Research in Computing Science 18, pp. 203-212. 2006

[15] Sherry, L. Issues in distance learning. International journal of Educational telecommunications, Vol. 1, No. 4, pp. 337-365. 1996

[16] Stenerson, j. System Analysis and Design for a Successful Distance Education Program Implementation. Journal of distance learning administration, Vol. 1, No. 2. 1998

[17] Taylor, J. Fifth Generation Distance Education, In the Proceedings of the 20th world Conference on Open learning and Distance Education, International Council for Open and Distance Learning, ICDE, Usseldorf, Germany. 2001

[18] Thorandike, R., Cunningham, G., and Hegen, E. Measurement and Evaluation in Psychology and Education. New York: Macmillan Publishing Company. 1991

[19] Wiersma, W., and Jurs, S. Educational Measurement & testing. Boston: Allyn and Bacon. 1991

Develop Case Studies to Teach Cryptography in A Collaborative Environment

Li Yang[1], Joseph Kizza[1], Andy Wang[2], and C. H. Chen[3]
[1]Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN, USA
[2]School of Computing and Software Engineering, Southern Polytechnic State University, Marietta, GA USA
[3]Computer Science Department, Tuskegee University, Tuskegee, AL USA

Abstract - *Universities which offer an information assurance degree, a certification or coursework have a course, or part of a course, devoted to cryptography. An effective teaching strategy that motivates and engages students in cryptography courses is necessary as we prepare IA professionals to meet the national needs. Currently, most cryptography course focuses primarily on the mathematical and theoretical aspect of cryptography. Therefore, there is a need to practice mathematical and theoretical aspects of cryptography. We designed a collection of case studies on cryptography which allowed students to gain experiences with cryptographic algorithms, especially increasing student awareness of possible threats and attacks to various cryptographic techniques. The impact of our pedagogical approaches has been assessed. Our effort contributes to cryptography and IA education in terms of case studies and IA education pedagogy.*

Keywords: A Maximum of 6 Keywords

1 Introduction with the need of studying cases using real-world scenarios

Information assurance (IA) has gained attention in multiple disciplines including Computer Science, Information Technology, Software Engineering, Information System, and Information Management. Teaching cryptography is essential in undergraduate information assurance (IA) education as it is an indispensable component in the IA knowledge domain and plays a key role in ensuring information confidentiality, integrity and availability.

At an event celebrating 30 years of public key cryptography in 2006, experts said that cryptography still lacked usability. Brian Snow, a retired technical director at the National Security Agency (NSA), notes that security products lack quality because they are poorly designed and often not in a secure way [6]. Cryptography in real-world networks and systems has not been as effective as they are supposed to be from the mathematical perspectives due to engineering challenges. Building real-world cryptographic systems is different from the abstract theories of cryptography with only pure mathematics. Designers and implementers face real-world constraints that often pose as engineering challenges. To prepare student for these real-world security goals, we must teach cryptography in real-world settings. Therefore, there is a need to design and incorporate case studies in a cryptography curriculum that simulate and capture real-world cryptographic applications. Case studies method in cryptography is a good approach to achieve this as it engages students in real-world settings, which inspires creativity of students and train them to adapt cryptographic solutions to emerging areas.

Our developed IA case studies tied with each cryptography topic teach students both cryptographic algorithm and vulnerabilities. The case studies also challenge students in a collaborative learning environment, which can provide a peer-supported environment and help students in IA overcome their apprehensions towards cryptography. Our approaches enrich cryptography and IA education resources, and provide a model of IA course ware development and teaching with cryptography as a core component.

2 Develop case studies to teach cryptography in a collaborative environment

We understand that extended periods of cryptography practices are essential in a variety of courses in the area of information assurance, but they often pose as a mental barrier for many students. We solved this dilemma by introducing peers into the learning process and establishing a collaborative learning environment, because learning from peers is a natural way of learning and students can better engage in active learning in a peer-assisted classroom than working alone [1].

We explored and developed case studies to exercise methods and principles of cryptography in real-world systems. Case study methods have been used in teaching science and engineering and shown benefits in instruction [2, 4, 5]. We used case studies that can help students delve into the specific security requirements and opponents in various emerging application areas and discuss the procedure about engineering cryptography into system design and implementation. This requires students to understand both real-world constraints of a specific area and to master cryptography algorithms and techniques. Students in a team built a secure system in real settings, which is the essence of cryptography in information assurance.

To help students with their thinking and enable them to design at high-levels, the instructors provided introduction and basic cryptography programming materials to students. The students used the building blocks to solve real-world cryptography problems after understanding specific cryptography requirements for each case. A collaborative learning environment was established based on "Collaborative Learning Techniques – A handbook for college faculty" [1]. A broad list of techniques in the Barkley handbook was explored, such as discussion group, reciprocal teaching, structured problem solving, and team matrix. One challenge in a group-based studying environment was to ensure individual accountability and to evaluate individual performance [3]. The instructor addressed these issues using the following methods: 1) in-class quiz, 2) participation of discussions, 3) regular shuffling of teams, and 4) peer reviews.

Students were required to write critiques for recently published cryptography papers with emphasis on understanding how to bridge real-world requirements with cryptographic techniques. Students were encouraged to form groups and select their own projects after discussion with the instructor.

We developed the cases in four areas: network security, E-commerce/Web security, security management, and digital forensics. Each case included a real-world scenario, problem analyses and activities, learning outcomes, and assessment.

Case 1: How do you secure BlackBerry devices?
Real-world Scenario: The BlackBerry is a wireless handheld device which supports push e-mail, mobile telephone, text messaging, internet faxing, web browsing and other wireless information services. It delivers information over the wireless data networks of mobile phone service companies.
Problem Analyses and Activities: We secure data traffic in transit between the BlackBerry Server and the BlackBerry devices and data stored in the BlackBerry device using cryptography. Three requirements specific to the BlackBerry were considered: 1) Communication is done through wireless links, which is more vulnerable than wired communication. 2) The limited computation capability in the BlackBerry. 3) The device may be stolen or grabbed by someone other than the owner.
Learning outcomes: Students gain knowledge of wireless handheld devices, and are able to take the above requirements into consideration when they design and implement cryptography techniques to protect data communication and storage.
Assessment: Students are able to secure wireless data communication as well as data storage.

Case 2: Do you trust others in virtual environment?
Real-world Scenario: Virtualization is almost a mainstream technology today because it reduces power usage, makes server and operating system (OS) deployments more flexible, and better uses storage and system resources. It helps to advance cloud computing, a new concept of collaboration and distribution. Virtualization technology allows one box hosts multiple virtual machines (VMs) and provides a computing environment for remote users. How do we provide separate VMs so that they cannot affect each other in one box? How do we ensure security, compliance and trust in a virtual environment?
Problem Analyses and Activities: 1) Investigate solutions to separate VMs in one box; 2) study how to provide authentication to virtual users; and 3) propose solutions to prevent data loss.
Learning outcomes: Students gain knowledge of virtualization technology as well challenges faced by virtualization. Students know how to separate VMs using cryptography techniques, and authenticate users in a virtual environment in a scalable fashion.
Assessment: Students are able to setup a virtual environment and implement proposed security solutions.

Case 3: Is SSL/TLS enough to secure e-commerce?
Real-world Scenario: E-commerce/Web services integrate web-based applications and allow communication among different sources.
Problem Analyses and Activities: 1) Design an authentication solution which enables more than three parities to authenticate one another. For example, a cardholder, an online merchant, and a bank are able to authenticate each other. Standards like SSL/TLS (Secure Socket Layer/Transport Level Security) only support point-to-point authentication. 2) Design an encryption solution that supports selective encryption. This is useful in a workflow scenario where a document may be processed by several applications, or signed, viewed, or approved by numerous people. Standards like S/MIME (Secure / Multipurpose Internet Mail Extensions) and PGP (Pretty Good Privacy) as well as SSL/TLS treat each message as a whole.
Learning Outcomes: Students are able to analyze requirements, design their own protocol to support multiple-party authentication, and selective encryption. Students learn and practice how to provide quality of protection through message integrity, message confidentiality, and single message authentication. The XML (Extensible Markup Language), web service security, SOAP (Simple Object Access Protocol) messaging were investigated.
Assessment: Students used written reports and oral presentation to demonstrate the results.

Case 4: How do you secure patient data?
Real-world Scenario: Health care information security planning, policies, digital right and privacy management are subject to Health Insurance Portability and Accountability Act

(HIPPA). Students designed their security policies and use cryptographic techniques to implement those policies.
Problem Analyses and Activities: 1) Students studied HIPPA and design security policies compliant to HIPPA. 2) They secured health data storage, electronic transaction and control access using authentication.
Learning outcomes: Students were able to design security policies according to regulations. They were able to implement security policies using cryptography techniques.
Assessment: Students wrote report and give presentations.

Case 5: Ensure the validity of forensic evidence by using a hash function
Real-world Scenario: A forensic hash is used for identification, verification, and authentication of file data. A forensic hash is the process of using a mathematical function and applying it to the collected data, which results in a hash value that is a unique identifier for the acquired (collected) data (similar to a DNA sequence or a fingerprint of the data). The National Child Victim Identification Program (NCVIP) creates hash sets to identify victim images of child sexual exploitation. Both MD5 and SHA-1 algorithms are commonly used on forensic image files. The hash process is normally used during acquisition of the evidence, during verification of the forensic image (duplicate of the evidence), and again at the end of the examination to ensure the integrity of the data and forensic processing.
Problem Analyses and Activities: 1) Students practice digital evidence acquisition by acquiring a forensic image, creating indexes, looking for evidence within the image, and generating reports. 2) Discuss collision in algorithm of MD5 and SHA-1. Since a hash collision would prevent known file filter identification, can criminals create a duplicate set of innocuous picture files whose hash values match a collection of contraband pornography?
Learning outcomes: Students understood hash algorithm and related cryptography methods used in law enforcement.
Assessment: Students wrote reports and present discussion results.

3 Evaluation of effectiveness

We used evaluation of students to assess effectiveness of case studies on cryptography education. Pre- and Post-course surveys were conducted at the beginning and end of each cryptography and IA course that adopted the case studies.

Survey on cryptography knowledge and interest of IA students:

1. Have you designed and implemented any cryptography related projects?

2. Have you used cryptography software for hands-on practice or case study?

3. Three key features of hash function?

Answers to the first two questions are on a 1-5 scale with 5 as the most favorable choice and 1 as the least favorable. Our evaluation shows that our solution improves student learning on cryptography, a fundamental component in IA.

Table 4.1 Hands-on exercises and case study in improving learning of cryptography
(The number of students in the survey is 23)

Questions	Pre-survey	Post-survey	Evaluation
Have you designed and implemented any cryptography related projects?	2.0 (mean)	4.5 (mean)	Improved
Percentage of students who used cryptography software for hands-on practice or case study	17%	100%	Improved
Knowledge question on hash function	4% right, 43% partially right, 52% wrong	75% right 25% wrong	Improved

4 References

[1] Barkley, E.F., Cross, K. P., and Major, C. H., Collaborative learning techniques – A handbook for college faculty. Jossey-Bass, 2005.

[2] Davis, C. and Wilcock, E. Teaching materials using case studies, available at:

[3] http://www.materials.ac.uk/guides/casestudies.asp

[4] Diamond, R.M., Designing and Assessing Courses and Curricula. John Wiley & Sons. 1998.

[5] Herreid, C. F. Using case studies to teach science, available at

[6] http://www.actionbioscience.org/education/herreid.html

Random Number Sequences Displayed in Two Dimensions

Jack Ryder
Computer Science, Kean University, Union, New Jersey, USA

Abstract - *The use of random numbers can be found in many applied areas of mathematics and computer science such as cryptography, gaming, statistics, software testing, simulation and modeling. One common method of generating pseudo-random numbers is the linear congruential method, which uses a modulus operation and eventually repeats the random number sequence. This repeating sequence of numbers forms a partition on the set of integers and we can graph the induced relation to show the corresponding cycle sequence. A method for dynamically displaying cycles in two dimensions generated by random numbers is presented.*

Keywords: Random Number Sequence, Visualization, Cycles

1 Introduction

The uses of random numbers in applied mathematics and computer science are many. Most humans have thought about, intuitively understand fundamental concepts, and enjoy the concept of randomness. Educators have the opportunity to introduce students early in their academic studies to random number generators. These generators produce streams of numbers, which eventually start to repeat, and are grouped into a cycle. To fully comprehend the nature of these random number streams and corresponding cycles it is helpful to visually display the results. This short paper briefly discusses random number generation, their corresponding cycle sequences, a method for visualizing random number cycles [1] and suggests where they may fit into the curriculum.

2 Random Numbers

Random numbers exhibit two very important properties: 1) uniformity in their probability density functions – equal likelihood of occurring, and 2) independence from previously generated numbers. Computer simulations use pseudo-random number generators that give the appearance of exhibiting these properties. They are typically implemented using a formula to generate the next random number based on the previous one, hence they are not independent and the use of the term pseudo. See [2] for a complete discussion of random numbers and their properties. The random numbers discussed herein are pseudo random numbers.

The linear congruential method for generating random numbers is widely used and given by the recurrence relation: $X_{n+1} = (aX_n + c) \bmod m$ where a, c, n, and m are positive integers, n starts at 1 and the first random number is choosen to begin (seed) the sequence. The longest sequence before the generator starts to repeat is called the period. The best generators have periods at, or close to, m. The quality of the random number sequence is highly dependent on the choice of a, c, m and the seed.

As an example of using the linear congruential method with the seed = 5, a = 5, c = 3 and m = 9, the random number sequence 5, 1, 8, 7, 2, 4, 5, 1, 8, 7 (repeating) is generated.

3 Cycles

A cycle can be characterized as a repeating path from node to node in graph theory. They can also be characterized as a permutation on a set of nodes. Consider the relation on the set of integers 0 to 8 given by: (0,0), (1,8), (2,4), (3,6), (4,5), (5,1), (6,3), (7,2), and (8,7). This relation partitions the set of integers into the following cycles: (0), (3,6), and (1, 8, 7, 2, 4, 5). This can be seen by the fact that 0 maps to 0, forming one cycle, 6 maps to 3 and 3 maps back to 6 to form the second cycle, and finally 1 to 8 to 7 to 2 to 4 to 5 and back to 1 forming the third cycle. Note the longest cycle in this relation is the same cycle given by the linear congruential random number generator in the preceding example.

4 Visualizing Cycles and Implementation

To visualize the cycle sequence of random numbers $r_1, r_2, \cdots, r_{n-1}, r_n$ we can plot them as a series of line segments in two dimensions using a sequence of ordered pairs $(r_1, r_2), (r_2, r_3), \cdots, (r_{n-1}, r_n)$. The sequence of ordered pairs when plotted becomes a random walk and returns to the starting ordered pair. For our example plotting the following ordered pairs in Table 1 gives the cycle displayed in two dimensions in Figure 1:

Table 1 - Random Number Sequence as Ordered Pairs

R(n-1)	R(n)
5	1

1	8
8	7
7	2
2	4
4	5
5	1

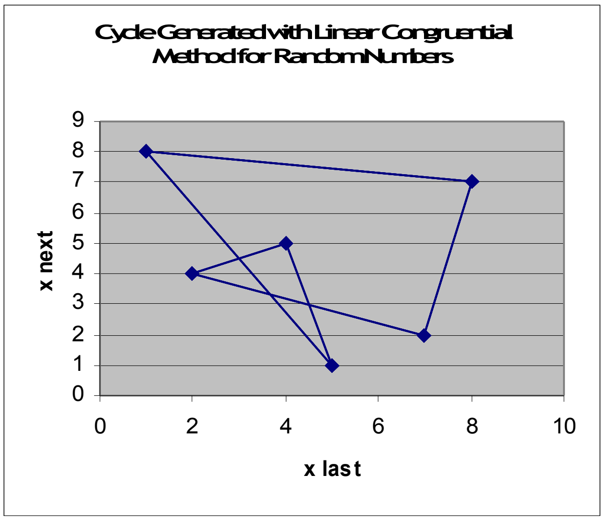


Figure 1 - Displaying Random Cycle Sequence

Since the linear congruential generator is required to repeat on or before m random numbers, the graph is guaranteed to display a cycle in its random walk. When using the linear congruential method to generate random integers between 0 and m there will be one or more cycles depending on the choice of a, c, m and the initial seed X_0. For example if all random numbers between 0 and m are in the cycle sequence there is one cycle, otherwise there is more than one cycle. Each cycle forms a "cyclic group" with related mathematical properties.

Implementation of the random number sequence, the ordered pairs and the corresponding display can easily be done using spreadsheet software. The steps are : 1) set up input cells for a, c, m and the initial seed, 2) calculate the sequence of random numbers in a column with each cell being a function of the previous random number, note the first cell will reference the seed and all others will reference the preceding number, 3) compute the second values of the ordered pairs as a column by referencing (copying) its corresponding random number, and 4) create a dynamic graph to display the two colums of ordered pairs as a series of line segments.

Students can investigate cycles generated by changing values of a, c, m and the initial seed. Any initial seed within a cycle will produce the same cycle graph. Checking all possible seeds for a specific random number generator will will display all possible cycles. It is easy to see that the cycles with length close to m will yield the maximum density and are the most useful as random number generators. Choosing prime numbers for a, c and m will show how they work better by producing longer random number sequences than multiples and divisors.

5 Final Thoughts on Use in Curriculum

The material on generating random number sequences and graphing cycles is applicable as introductory/motivational material for beginning students as well as for those interested in further study. Computer science students can explore underlying concepts by automating (programming) procedures to identify cycles, partitions and investigate properties of built in random number generators. Students in modeling and simulation courses will be interested in testing for quality (uniformity and independence) before using/selecting a random number generator. Discrete mathematics and algorithm courses can use these concepts to investigate graphs, partitioning, connectivity, reachability and transitive closure.

6 References

[1]J. Ryder, "Visualizing Cycles in Random Sequences", *Mathfest 2009 Proceedings* (Mathematical Association of America), August 2009

[2]J. Banks, J. Carson, B. Nelson, D. Nicol, *Discrete-Event System Simulation*, Pearson-Prentice Hall Publishing, 2005

A Model of Interdisciplinary Curricular Collaboration: Inquiry Based Real-World Case Studies for Introductory Bioinformatics and Computer Science Courses

Vasiliy Kolchenko[1], Tatiana Malyuta[2], Raffael Guidone[2]
[1] Department of Biological Sciences, New York City College of Technology, Brooklyn, NY 11201 USA
[2] Department of Computer Systems Technology, New York City College of Technology, Brooklyn, NY

Abstract – *An interdisciplinary team of computer science and biology faculty is developing inquiry based real-world case studies for introductory bioinformatics and computer science courses. The collaboration was designed to provide a model of such work and explore its challenges, benefits and potential. The step-by-step process of curricular development is analyzed. It included the selection of a real-world science problem, the integration of the project into existing curriculum and the creation of a wide range of corresponding learning activities. Combining scientific inquiry, service learning and computer science tools and methodology is a promising pathway for bringing undergraduate science, technology, engineering and mathematics (STEM) education into the twenty first century. The guidelines for future partnerships are discussed.*

Keywords: interdisciplinary curriculum, bioinformatics, real-world problems.

1 Introduction

Interdisciplinary inquiry based projects have become the latest frontier in STEM undergraduate education [1]. However, they need to represent more than the sum of the individual STEM fields [2]. Integration of diverse subjects in the real-world setting creates a challenge for faculty whose expertise is limited to a specific discipline. In undergraduate courses, it requires a productive collaboration of instructors from different departments and schools, which is often lacking. Particularly important is the dialogue between specialists in computer science and natural sciences (biology, chemistry and physics). This partnership is becoming critical in STEM research as evidenced by the growing role of bioinformatics and computational biology, physics and chemistry. At the same time, undergraduate STEM education curricula are often lagging behind in adopting the changes. Although there are instances where the collaboration of instructors from different disciplines has produced effective curricular modules, institutional barriers and differences in training, experience and professional cultures often have a limiting effect. To optimize the effectiveness of these interdisciplinary collaborations and the curricular products they produce, we conducted a case study of interdisciplinary curricular development that was supported by NSF CPATH and NSF I^3 awards at New York City College of Technology. Faculty from Computer Systems Technology (CST) and Biological Sciences Departments developed curricular units that could be utilized in introductory bioinformatics and computer science courses. The goal of the study was to explore the challenges of interdisciplinary curricular collaboration and provide the guidelines for future partnerships. The pedagogical goal was to introduce real-world inquiry based interdisciplinary class projects into introductory STEM courses in order to engage students in active learning and problem solving. The learning activities aim to give students a taste of actual field projects as they are developed in industry based on the computer system analysis and project management methodology [3, 4].

2 Project design process

2.1 Selecting a real-world STEM problem

The real-world inquiry based projects that integrate computer science and biology may address a variety of current STEM research problems and use a number of applications. As we discussed the options, the following criteria for selection were considered.

1. Will the project allow the students to relate to the problem on the personal level? Are there familiar themes and experiences providing additional motivation for the learning activities and engaging student imagination, sense of service to the community and scientific curiosity?
2. Are there fundamental biology and computer science topics that could be taught based on the project? How do these topics fit into the course curriculum? How can the traditional teaching of the subject support the project and vice versa?
3. Does the level of the required learning activities correspond to the introductory STEM course? How will the project complement the course instruction? Will every student in the class benefit from the project?

Among the potential choices in biology were the following: development of a new drug in the pharmaceutical industry, research of a particular genetic disease and epidemiological study of an infection outbreak. All three satisfied the first criterion but elicited various considerations in regard to the second and the third. After extensive discussion, it was decided that a particular genetic disease study will provide the optimal balance of the appropriate level of complexity and the wide range of topics required in the introductory bioinformatics course. The choice of the genetic condition was *sickle cell disease*. It is a well-known affliction that combines a diverse background in molecular biology and genetics, a dramatic history of advanced research and a number of unresolved medical problems [5, 6]. Addressing some of these problems is the dual task of bioinformatics and computer science. The need for databases, computer tools and online applications in these combined fields is great and presents an excellent opportunity for the real-world projects.

2.2 Integrating the project into existing curriculum

The curricular components used in the case study were extracted from three courses: Elements of Bioinformatics, BIO 3350, Systems Analysis and Design, CST 2406, and Computer Information Systems Project Management, CST 4800. The project instructional materials could be modified for implementation in any of these courses, at the appropriate level of detail. The contributions of these curricula complement each other. Bioinformatics provides scientific framework for addressing real-world biological and medical problems. It is based on the modern knowledge of molecular biology, particularly on understanding of DNA sequence, gene expression and protein sequence, structure and function. The framework includes using resources and methods of computer science and information technology – computer algorithms, databases, search tools and sequence alignment tools, among others. *Sickle cell disease* provides an excellent model of a genetic condition. It is caused by a substitution mutation in a single gene coding for a major protein in the red blood cells – hemoglobin. One letter «misspelling» in the gene's DNA sequence leads to alterations in the protein sequence, structure and function. This, in turn, causes sickling of the red blood cell and a multiplicity of signs and symptoms. Although the pathogenesis of the disease is well understood, the treatment options are limited and the condition may be fatal. Partial immunity to malaria in carriers explains high prevalence of the disease in the tropical countries. Potential topics for discussions include population genetics, poor choices for disease prevention (family counseling and planning), search for better treatments and the prospects for finding the cure.

The curricula of two computer courses, Systems Analysis and Project Management, contribute general methodologies for the development of information systems and for executing the phases of the project – the sequence of steps and milestones, the format of documents and products designed in the process of development of an information system or computer application for real-world problem solving. The students in the Bioinformatics course will benefit from this systematic approach, which is often omitted in the bioinformatics curriculum. It can be considered as the scenario of the professional internship: this is how computer science specialists create and manage information systems in response to user requests. This is a central feature of the case study that makes it applicable to other STEM courses and projects. The methodologies will remain, but specific bioinformatics-related tasks will be substituted by other discipline related tasks. The students in the two computer courses will benefit from working on a science oriented project which brings a new prospective to the course – excitement of scientific discovery, urgency and humanitarian significance of medical research, the potential benefits to the effected population. The project is done over the course of the semester, in parallel with regular topics, which complement the activities. In the Bioinformatics course, the greater emphasis is on the scientific background and subject specific tools. In Systems Analysis course, it is on the process and products. To formalize this «cross-pollination» process and structure the curricular integration and its assessment, the following guiding questions were proposed.

1. What are the curricular contributions of each field to the project? Are there synergies between them that add value and bring the conversation in the classroom to a higher level?
2. What are the benefits of the project to students in each course? What kind of new prospective, knowledge and skills does it bring to the students?
3. What are the time demands of the project activities vs. existing curriculum? What is the optimal time line and time management for the project over the course of the semester?

2.3 Designing learning activities

The answers to the questions about time allotment require considering project design and learning activities in greater detail. However, an inquiry based case study does not start with clearly defined instructions that predetermine all student actions. It begins with a scenario or question that requires active involvement of students in designing the potential solutions [7]. This approach driven by project management methodology may be suggested for a number of STEM disciplines. Its general value is in providing a framework for interdisciplinary collaboration that can be applied in a variety of the curricular development case studies. In the context of the sickle cell disease research that served as a unifying theme for all the learning activities in this particular case study, the following scenarios were suggested.

1. Understanding data required for researching various aspects of a particular biomedical problem. Choosing a problem, e.g. researching genetic variants of the gene associated with the sickle cell disease and gene expression patterns, and planning to develop a

database. Participants: students of the bioinformatics class.

2. Defining the scope of work for the database development for the chosen problem, e.g. sickle cell disease protein sequence, structure and function in relationships with the gene expression patterns in different patients. Participants: students of the bioinformatics and computer science classes.
3. Defining and refining system requirements for the repository of gene variants and gene expression data in sickle cell disease patients. Participants: students of the bioinformatics and computer science classes.
4. Modeling system requirements for the repository of gene variants and gene expression data in sickle cell disease patients. Participants: students of the computer science and bioinformatics classes.
5. Designing and developing a simple database [8, 9] for the repository of gene variants and gene expression data in one of the available software/tools (MS Access, MySQL or commercial database management systems). Participants: students of the computer science classes with the assistance of the bioinformatics students.
6. Populating the database with data and executing simple queries that may illustrate and define the benefits of the database for research of the problem. Participants: students of the computer science and bioinformatics classes.
7. The project can be continued in subsequent interdisciplinary courses, e.g., elective project based courses in computer science and other STEM disciplines.
    a. Developing a basic web application for the online sickle cell disease research database (with computer science students).
    b. Developing a set of analytics to process data from the sickle cell disease research database in order to analyze the trends and tendencies identified in the data (with applied mathematics students).

The projects and activities may be rotated in different classes and semesters. All of them will require a general introduction to the sickle cell disease, its biological background and societal implications. Systems analysis introduction will include case background, organization structure, information systems facilities and problem description. Better understanding of the systems analysis process and stages help streamline and guide the steps of the STEM problem solving in the classroom, as it does in the real world scientific exploration and discovery. These phases include scope definition, problem analysis, requirements analysis, logical design and decision analysis [3]. The projects based on designing particular tools (databases, web servers) will require more technical skills and can be assigned to more advanced and experienced students. Other projects involve systems analysis without designing the actual system. They can be successfully implemented by all students working in small groups over the period of a few weeks. In Bioinformatics, the major topics of the course (DNA sequence analysis, protein sequence, structure and function, gene expression analysis) are illustrated and reinforced by the *sickle cell disease* case study. The projects are completed and summarized in group presentations to the class by the end of the semester.

3 Challenges, benefits and the potential of the interdisciplinary collaboration in undergraduate computer science and other STEM courses

When faculty from different departments and schools (School of Technology and Design and School of Arts and Sciences) start collaborating on the interdisciplinary curricular STEM projects, the typical initial challenge is not knowing well enough what the other is doing, both in the classroom and professionally, in industry and research. This mimics the initial situation between computer science specialists and users in multiple fields. It is necessary to overcome this obstacle by learning from each other and from the literature, and the earlier it is done, the more productive the collaboration becomes. In our experience, participation of the external partner from the Polytechnic Institute of New York University provided an additional impulse in the beginning of the project. As the Director of the Bioinformatics program, he had both the academic and the industry perspective, and helped to launch the conversation. Exchanging curricular materials and discussing them regularly in informal meetings were essential. The benefits of learning the principles, the language, the goals and the techniques of the partnering discipline extend beyond one project. This understanding will inform one’s teaching and add new aspects to the instructor’s view of science and technology. The next challenge was to make the optimal selection of the real-world STEM problem for the case study. Here, the necessary step is looking at the problems from the students’ point of view. It is important to select an exciting challenge that combines the potential for scientific inquiry at the appropriate level, ideas of service learning and the necessity of using computer science tools and methodology. Although we do not include service to the community as a project requirement, the premise of helping people in need is an important motivational component that can be emphasized, implicitly or explicitly, in a variety of ways [10]. The scientific inquiry and computer science components should be tailored to the student academic level. In this respect, expecting too much or too little from the students may cause unanticipated problems and interfere with the project progress. The benefits of good problem selection are significant, both in motivating the students and in allowing them to achieve success in active learning. Integrating the project into the existing curriculum creates two basic challenges. One is the correspondence between the project and the fundamental topics of the course. In the case study, the project does not replace the topics but complements and reinforces them. Another challenge is the time constraints that any existing course curriculum imposes. Introducing a new project or learning activity requires additional time – both instructional time in class and individual and group work time either in class, at home or at some other facility. Balancing it and improving student productivity is the key to success. It provides the students with new perspective, additional skills and a more meaningful learning experience. Finally, learning activities of the case study, in addition to being highly technical, need to engage students' imagination in the context of the real-world professional situation. There are two basic roles they can play in these scenarios: one is computer science specialists developing the computer system and another is product users or clients, e.g. biologists, medical doctors or biotech researchers [11]. In our classes, the first role most likely belongs to the computer science and applied mathematics majors and the second – to the majors in other science and technology programs. The level of student engagement in the activities depends on many factors including clarity of introduction, optimal level of assignment difficulty and fair assessment, among others. The assignments are not limited to the ones proposed above. These are just a few examples that could be implemented and continuously modified and improved based on the feedback from the students. An interdisciplinary collaboration in undergraduate computer science and other STEM courses has the great potential for new curricular module development, primarily because of the interdisciplinary nature of modern science and technology, and it is likely to proceed along the lines of the proposed model.

4 Conclusions

This case study applied a systematic approach to exploring challenges and benefits of the interdisciplinary curricular collaborations in computer science and other STEM courses. Among the conclusions was the realization that, in spite of considerable difficulties, especially in the initial stage of the study, such collaboration is possible and even vital for the future of the undergraduate STEM education. The project worked on two levels. First, it was beneficial for faculty from different departments and schools to initiate a productive conversation about the interdisciplinary curriculum. This faculty professional development case study may serve as a model of collaboration that can be easily adapted in other disciplines. The step-by-step approach, the guiding questions and criteria and the guidelines for future partnerships provide the framework for future curricular work. Second, the proposed modules provide an opportunity for the students in computer science and bioinformatics courses to engage in interdisciplinary collaboration which is typical for modern science and technology, both in industry and in research. At City Tech, a series of similar case studies has been in development including faculty collaborations between the Electrical Engineering and Physics Departments and between the Architectural Technology and Mathematics Departments. Administrative support and encouragement of these initiatives appear to be a critical component of success.

5 Acknowledgements

The work reported in this paper was partially funded by the National Science Foundation, NSF CPATH award # 0939120 and NSF I^{3}award # 0930242. Prof. Kalle Levon, Director of the Bioinformatics program at the Polytechnic Institute of New York University, contributed to the discussion of the project design.

6 References

[1] Maria A. Ruiz-Primo, Derek Briggs, Heidi Iverson, Robert Talbot and Lorrie A. Shepard. "Impact of Undergraduate Science Course Innovations on Learning"; Science (AAAS), Vol. 331, 1269-1270, March 2011.

[2] David J. Asai. "Measuring Student Development"; Science (AAAS), Vol. 332, 895, May 2011.

[3] Jeffrey Whitten and Lonnie Bentley. "Introduction to Systems Analysis and Design" McGraw Hill Publishing Company, 2006.

[4] Kathy Schwalbe. "Information Technology Project Management". Course Technology Publishing Company, 2009.

[5] Jonathan Pevsner. "Bioinformatics and Functional Genomics". John Wiley & Sons Publishing Company, 2009.

[6] Betty S. Pace. "Renaissance of Sickle Cell Disease Research in the Genome Era". World Scientific Publishing Company, 2007.

[7] Bruce Albert. "A New College *Science* Prize"; Science (AAAS), Vol. 331, 10, Jan. 2011.

[8] Graeme Simsion, Graham Witt and Morgan Kaufmann. "Data Modeling Essentials". Morgan Kaufmann Publishing Company, 2004.

[9] Len Sinverston and Paul Agnew. "The Data Model Resource Book, Vol. 3: Universal Patterns for Data Modeling", Wiley Publishing Company, 2009.

[10] Robert G. Bringle and Julie A. Hatcher. "Implementing Service Learning in Higher Education", The Journal of Higher Education, Vol. 67, No. 2, 221-239, Mar. 1996.

[11] Bryan Bergeron. "Bioinformatics Computing". Prentice Hall Publishing Company, 2003.

Evaluation on Teaching Embedded System with Portable Labs in a Box

Kai Qian*
Computer Science
Southern Polytechnic State University
kqian@spsu.edu

Xiaolin Hu
Computer Science
Georgia State University
xhu@cs.gsu.edu

Liang Hong
Department of Electrical and Computer Engineering
Tennessee State University
lhong@tnstate.edu

ABSTRACT
The rapid growth of embedded systems results in a shortage of professionals for embedded software development. However, embedded system education is not well represented in current Computer Science (CS) academic programs. Due to budget cuts and resource limit, it becomes harder for higher education institutions to maintain traditional laboratory than ever before. In this paper we present our implementation of embedded system courseware with a carry-on lab in a box (a very affordable 8051 microcontroller development kit) which can be conducted anywhere and at any time. It can be implemented on line, partial on line, or in class room. Its portability allows students to work on the labs at anywhere anytime. The evaluation has shown that hands-on lab oriented curricula well engage students in learning and prepare our graduates directly into the nation's workforce.

Keywords
Embedded, portable, real time, engineering

1. INTRODUCTION
Many schools realize that we need to enhance embedded systems education to meet the rapidly growing demands in industry. Unfortunately, existing laboratories generally require significant investment in resources. Due to limited resources including budget, equipment, lab space, or faculty expertise, many schools teach such courses without hands-on lab practices or with antiquated equipment which drastically hinders student learning. Existing embedded systems instruction relies on significant investment in lab resources and high requirement for instructor's expertise. To overcome these difficulties, we have developed an online embedded software courseware with hands-on labs in an inexpensive portable box. The portable and modular design of this courseware provides a "ready-to-adopt" model for broadening embedded system education. This courseware is especially suitable for the universities/colleges that have the need of embedded system education but are constrained by limited financial budget, scarce dedicated staff and faculty and lab facilities.

2. MODULAR AND PORTABLE LABWARE
The developed courseware adopts a modular structure that organizes the courseware into a sequence of teaching modules. Each module includes lecture notes, PPTs, review questions, hands-on laboratory practices, and assignments. It also provides portable real labs based on a popular MCU development kit. All learning materials and lab manual videos are posted online. Because of the modular design, the courseware is easy to be integrated in other undergraduate computing courses. Meanwhile, the hardware needed to teach these modules is minimized due to the portable labs. We have used these modules in an online embedded software development course and a CS capstone course. We plan to integrate the course in other courses such as software engineering, computer organization, and operating systems. All hands-on labs are developed and implemented using the portable, supercharged 8051 MCU based C8051F005DK development kit from Silicon Laboratories Inc. which comes with necessary I/O, serial ports, and a basic RTOS. It costs at an average textbook price. The kit comes with an Integrated Keil C51 IDE which provides students with the hand-on opportunity to work with real-world embedded system projects instead of simulation only.
This course is designed with a particular focus on the software design and development of embedded system rather than the hardware design although students will be exposed to hardware design especially in the I/O peripheral layouts and interaction controls.

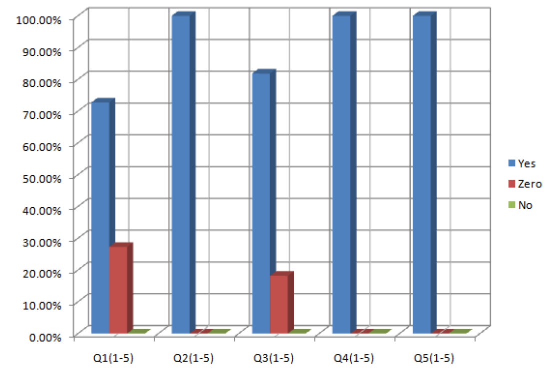
Module	Contents
Introduction to embedded system	Overviews of embedded system concepts, fundamentals, Components, Constraints
Embedded system software engineering	Time Constraint Analysis with RMS and EDF Modeling and architecture (Context Diagram,

	DFD, FSM, StateChart) Design criteria
Microcontrollers 8051	Microcontroller,I/OPorts, , mory, SFR, Interrupts
Embedded programming	C51, Assembly for 805
Concurrency	RTOS based multi-tasking scheduling, Round-robin Preemptive priority, Sync and Async tasking
Serial communications	Communication modes, UART, serial Port I/O, Handshaking, Baud Rate
Case study and Projects	Software engineering SDLC

The course is planned in such a way that students get hands-on practice along with the theory which boosts the students' confidence by working on the real hardware and software platforms to take on challenges of the real world. The hands-on labs focus mainly on applying all the embedded system concepts such as RTOS on the C8051F005DK Students have hands-on experience to model, analysis, design, development, and testing embedded software with real microcontroller rather than simulations.
Hands-on labs with kits include 3 digit 7-segment LED counter, Digital clock, Temperature control system with sensor, keypad, and LCD display, RS232 with PC, Traffic RTOS control system.

3. EVALUATION

This class becomes one of the most popular elective classes in CS and SE curriculum since we offered it. The student feedback is always overwhelming positive. Many students said in the class survey that they had great opportunities to practice the theory and abstraction to practical application; to integrate their knowledge learn in the CS and SE programs to solve real-world problems; to co-design and develop software with microcontrollers to produce a real world embedded systems on their own. Students enjoyed in working with development embedded software in cross-platform environment. One of the contributions of the designed labs and equipments is the inexpensive, completeness, and feasibility for any schools with limited financial recourses. It also provides a model for any instructors to start up an embedded software course with supported labs. The following chart shows the feedback from students.



Q1. I like this PEARL online courseware
Q2. The labs in a box provides hands-on experience on embedded system design.
Q3. The online instructional materials help me understand concepts better.
Q4. The online lab guidelines (video, demo) help me work on labs and assignments
Q5. The textbook has a good fit to the courseware.

There are five score survey choices for each evaluation question:
Strongly agree(5), Agree(4), Neutral(3), Disagree(2), Strongly Disagree(1).
The YesPercentage bar represents the choices of "Strongly agree" and "Agree"; The NoPercentage bar represents the choices of "Strongly Disagree" and "Disagree"; The ZeroPercentage bar represents the choices of "Neutral". The student survey chart shows a very positive learning assessment.

4. CONCLSION

This paper presents a new pedagogical teaching and learning model for improving student learning in embedded system and preparing students for tomorrow's embedded system workforce. Most students enjoyed what they learned in the new courseware. In particular, students favor the hands-on portable labs. Students were excited with their creativity opportunity of working on the embedded projects with the kit.

ACKNOWLEDGMENT

The work is partially supported by the National Science Foundation under award: NSF CCLI grant award #0942097 and #0942140, Portable, Modular, Modern Technology Infused Courseware for Broader Embedded Systems Education. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation

Saving Computer Science: Why, Where, and How

Scott A. Burgess
sab43@humboldt.edu
Department of Computing Science
Humboldt State University
Arcata, CA 95521

Abstract

Computer science programs in the United States will likely be under scrutiny for several years due to insufficient funding for higher education. This paper examines why computer science is a convenient target for budget cutting, which programs are most likely to be targeted, what the long term impact is likely to be, and how departments can be saved. Our own computer science program at Humboldt State University provides an example of a department saved through retooling of our academic business.

Keywords: Program elimination, enrollment, budget cuts, prioritization.

1. Introduction

While the U.S. economy has achieved some measure of stabilization, it has begun to dawn on some that the next period of significant, steady growth may be many years away [1,2,3]. Further, if Koo's primary hypothesis about balance sheet depressions is correct a strong reduction in government spending could still unleash a depression in the near future. But economic catastrophe notwithstanding, the balance sheets of most state governments are in poor condition and spending increases appear unlikely for the foreseeable future. Given states' increasing long-term fiscal obligations and the apparent public desire to address fiscal imbalances through spending cuts, decreases in actual outlays toward higher education instruction appear likely [4].

At many campuses nationwide, reductions in funding have led to program elimination. In the following sections I examine why computer science is a convenient target of budget cuts, where this is most likely to occur, and how to deal with the prospect of elimination. Our department, which faced elimination during the past year, is used as an example throughout.

2. Why cut computer science?

Computer science on our campus became a target for three specific reasons which are generalizable to other programs. In this, one should understand that we are a campus in the California State University system, and thus our campus resembles many state funded small comprehensive universities around the country. There undoubtedly are other reasons beyond the three I explore, but I am constraining this discussion to topics of greatest relevance to other departments.

First, our department has experienced great reductions in enrollments. As exemplified by the 2009-2010 Taulbee Survey [5], newly declared enrollments in computer science and computer engineering are only now beginning a modest recovery from their drop between 2002 and 2005. Newly declared enrollments dropped almost fifty percent during that period, while Bachelor of Science degree production dropped just over fifty percent between 2004 and 2009. The most immediate consequence of this is that many departments became overstaffed.

At the start of this period, our computer information systems program alone enrolled more than 200 students, with approximately 50-75 freshman enrolling every year. Additional faculty members were hired through summer of 2000, bringing tenure track faculty to eight. We added a computer science degree in 2002, but enrollments for both programs dropped to approximately 80 total students. From an administrative perspective, we went from one of the most cost effective departments on campus to one of the least cost effective.

Additionally, our campus suffers from a remoteness problem that affects only a small percentage of public campuses in California: we recruit the majority of our students from an average of more than 200 miles away. So while a rising tide does indeed raise our boat, we lack the local student base to retain enrollments when the tide suddenly recedes. We believe this effect exacerbated enrollment problems for departments on our campus.

Second, computer science remains an ugly stepchild of the university family. As a relatively young discipline, computer science is not well understood by other faculty. This effect may be complicated by faculty and staff who believe that they are doing computer science already because they use computers daily in their work, for everything from word processing to analysis of experimental data. Even discipline identity remains a constant subject for debate among members of our field, and computer science programs are variously grouped with campus engineering programs, science programs, or a nebulous "something else" depending on which campus one visits.

The youth effect also means that administrations seldom embrace former computer science faculty among their senior ranks. When our program was first discussed for possible elimination from our College of Natural Resources and Sciences, the campus president, our dean, and our associate dean were all biologists. There was no natural advocate in the administration to defend the program.

Third, our department comprised several faculty judged by administrators to be weak and/or problem faculty. I have spoken with faculty from departments across the West Coast, and I believe several aspects of this may be common to smaller computer science departments in our region.

This perceived weakness may again be partly attributed to the youthfulness of our discipline. Newer departments generally seem to have greater teaching expectations and lower research expectations, and faculty in our fast-moving field appear to spend more time staying current in the discipline, but still administrators often perceive departments with greater publication-to-faculty ratios as "healthier."

Publications in computer science are viewed differently outside our discipline. While conferences in computer science (eg: OOPSLA) may be considered equal to or better than journals within their subfields, most administrators from other scientific and engineering disciplines give greater weight to journals and this may color their view of the relative strength of computer science programs.

Small computer science programs must overcome several difficulties in recruiting faculty. Wages in many universities are simply not competitive with private employment for computer scientists. Universities do not have the market cornered on research in computer science, the way they do for some other disciplines. Small programs are numerous and widely distributed—but their locations are often in smaller cities far from the urban centers which act as magnets for computing companies, and therefore culturally far from the technology-proficient relationships that many computer scientists crave. Given these challenges, departments may genuinely struggle to hire the best quality faculty.

Our department additionally exhibited several faculty members whose behaviors and attitudes were viewed by many as, in a word, *unprofessional*. As chance would have it, they were also approaching retirement.

3. Where are the dangers?

Departments in universities may have many structures, programs, and capabilities for transformation. It is potentially useful to characterize departments which are more likely to face elimination.

Departments achieving sufficient maturity to support graduate programs are probably among the least likely to face total elimination. This should not be taken to mean that they could not lose a program or be required to restructure—some faculty have reported that their campuses are seeking to pare down expensive graduate programs since they are not necessarily as profitable as undergraduate programs, and because graduate programs seldom exhibit economies of scale—theses and projects are still handles on a student-by-student basis.

Among departments with strictly undergraduate programs, large programs may face difficulties if their student population drops precipitously, but restructuring may be sufficient to avoid elimination. And the more faculty members on tenure track lines, the higher typically the probability is that some are near retirement. Thus attrition may more easily reduce costs from faculty lines. Also, larger departments may employ some economies of scale with temporary faculty. In the recession of the early 1990s, the CSU campuses frequently resorted to layoffs of contingent faculty to achieve temporary "savings." The difficulty, as noted in the introduction, is that the current economic malaise is unlikely to be a transitory phenomenon.

Smaller computer science departments present the most worrisome circumstances. With few economies of scale to be exploited, there are often few ways to reduce expenses. Enrollment drops in computer science have been on average larger than most programs. And given the hiring burst which occurred shortly before the dot.com bubble broke, these programs are more likely to be overstaffed.

Why should we be concerned? Anecdotally, there has been mention that half of all computer science majors graduate from smaller departments. While I have not found a publication affirming this directly, cross-referencing the Taulbee Survey with the 2010 Census [7] does provide some support for this hypothesis: of the 265 Ph.D. granting departments contacted for the Taulbee Survey, 195 departments responded. Those departments account for approximately 11,500 computer science and information systems graduates for 2010. The total number of B.S. degrees awarded as reported in Table 298 of the Census for 2008 is approximately 42,000. It seems reasonable to conclude that after adjusting for the remaining doctoral programs, that with high probability at least half of all majors graduate from comprehensive and liberal arts programs. Projections are that computer science programs will likely graduate less than one third the needed number of software specialists [8]. This means that saving small departments an important task for our industry and our country's economy.

4. How to save a department

Departments facing elimination will need faculty to directly address the most pressing concerns of their campuses. These concerns may vary, but will often revolve around student enrollments and the campus administration's assessment of the relative health of the department.

Addressing enrollment concerns needs to be the first order of business. In our experience at Humboldt State University, the single most useful argument on behalf of the department was a graph obtained from Calvin College [8]. The graph illustrates that for the foreseeable future, almost 3 out of every 4 jobs created in science, engineering, technology and mathematics will be in computer science. Additional data from the National Association of Colleges and Employers showed that wages were healthy in the industry. These numbers combined to show an industry likely to recover jobs and exhibit strong wage growth, both likely to be attractive to prospective majors during a protracted recession.

While the supply is therefore evident, we also had to exhibit a plan to tap that supply which a reasonable critic might believe had good odds for success. Our plan opted to drop the computer information systems degree, retaining only the computer science degree. As faculty members were already retiring, this meant we could hold faculty costs down while still offering a degree. The remaining computer science degree was also significantly restructured—it included those elements of both degrees which had past recruiting success, but also featured new elements based on popular elective courses. Our internship program has been expanded, which we hope will attract prospects. We have added a Bioinformatics certificate program in cooperation with our biology department. And we are integrating an advertising program and new web site to draw attention to our new program.

We addressed the second concern by connecting with IT staff on campus who could give us feedback on our graduates (who constitute almost half of the IT staff), and establish some indirect connectivity to the campus administration. This has had minimal benefit to our department, with most of the value coming from feedback and guidance about our program structure.

Remaining faculty members have made some efforts to demonstrate competency and professionalism. We have recently filed a grant application to receive a LittleFe mini-supercomputer. We likely will seek external funds to refresh our Internet Teaching Laboratory. And we are actively participating in workshops and conferences.

We are not in the clear, of course. Proving we have done more than rearrange deck chairs on a sinking ship is a Titanic task. We expect to be at it for years.

A final worry arises from observations among mathematicians that recent growth in majors has been primarily among the largest institutions [6]. Data from the Taulbee Survey shows relatively flat enrollments for the largest CS degree granting institutions, but the incompleteness of that data does little to resolve the issue. It may be that larger departments are better at marketing their undergraduate programs by tying them to internship experiences and research experiences. This avenue deserves further investigation.

5. Conclusions

Our computer science department at Humboldt State recently received probationary status. That status requires us to improve enrollments with existing faculty to approximately 80 full time students over the next three years. We believe this is challenging but doable. This is a remarkable result for a department evaluated by the administration as the department most deserving of elimination, with the two programs (computer science and computer information systems) ranked most deserving of elimination. The nursing department at our institution, ranked second, was not so fortunate and has been eliminated on our campus. Two other programs, German and Applied Technology, have also been eliminated. No other programs reaching a similar status on our campus have been saved so far.

Other computer science departments should evaluate themselves with respect to the reasons computer science programs may be cut. Are your enrollments sufficient to sustain your current faculty numbers? Do you have functionally strong communications with your campus administrators? Are your faculty demonstrating competency? With respect to faculty competency, in particular, it may be best to illustrate this through actions rather than words, with emphasis on publications and grants.

Departments which face elimination may look to this article for ideas about how to address their predicament. While we are a remote campus, departments near to and carrying strong relationships with industry may find friends there who are reliable allies in the fight against elimination.

References

[1] Talbott, John R. *Sell Now!: The End of the Housing Bubble*. New York: St. Martins Griffin, 2006.

[2] Koo, Richard. *The Holy Grail of Macroeconomics*, *Revised Edition*. New York: Norton, 2009.

[3] Talbott, John R. *Contagion*. New York: Wiley, 2009.

[4] “Crisis of Confidence Threatens Colleges”, in *The Chronicle of Higher Education*. May 15, 2011. http://chronicle.com/article/A-Crisis-of-Confidence/127530/

[5] Zueben, Stuart. “2009-2010 Taulbee Survey”, in *Computing Research News* **23** (3), May 2011, p. 6-21.

[6] Bressoud, David. “Disturbing Trends in the CBMS Data,” in the web site of the Mathematical Association of America: http://www.maa.org/columns/launchings/launchings_06_08.html.

[7] The 2011 Statistical Abstract. United States Census Bureau, 2011. http://www.census.gov/compendia/statab/2011/tables/11s0298.pdf

[8] “The Market for Computing Careers,” at the web site of Calvin College, of Grand Rapids, Michigan: http://cs.calvin.edu/p/ComputingCareersMarket.