Abstract

There are some programming obstacles that the students have to deal when they are trying to solve a problem using a programming language, like the inexperience of the language, the unknowledge of some algorithms, and some cases the amateurishness of how to test a solution. The technological tools for developing programs and computer systems heavily explore the syntax and semantics of a program but do not provide support to determine its correctness and this is where students may failed by not exploring the correct scenarios and test cases. In this paper, we present the use of automatic evaluation tools as a way to help to remove these obstacles and improve learning experience, coding and testing skills in students.

Keywords: automatic evaluation, programming skills, problem solving.

1. Introduction

In order to develop strong programming skills, it is critical for students to be constantly challenged by task and problems involving computational solutions [6]. Generally students face a large number of these during the semester [4] so they can build and acquire the necessary skills and techniques needed in program development [1].

To do their jobs, students use development tools for writing code and building computer systems. This set of tools reduce code related issues and strongly explore syntax and semantics of a program, but do not provide support in the analysis of the solution of the problem, this being a major drawback since, most students verify their solution with few cases -usually the most basic to the problem in question- and therefore produce incorrect or incomplete solutions [5].

Some of these exercises are made inside of the classroom where the teacher could guide to the students, and outside of the classroom the student made them without any support or help, and they do not could know how their solution is.

In this paper we present how automatic evaluation software can provide teachers and students a way to improve the programming learning experience, coding and skills, and how a tool like this can complement the problem-solving skills to students, and break programming obstacles inside and outside of the classroom.

2. Background

As part of the programming courses professors suggest using one of the multiple development tools on the market such as Eclipse, Visual Studio, XCode, among others, as they rely heavily on the development of programs from the perspective syntax and semantics in addition to which may be employed for programs in multiple languages.

For several semesters we have seen that most of the solutions of our students are not entirely correct, as they usually poorly tested solutions by only working with the basic scenarios and not exploring the results on corner cases.

Based on this fact, we researched on platforms that support the evaluation of the solutions given by students to their assigned problems. The tools found provided a set of problems that students could solve in several programming languages and validate their solutions by running well designed test cases. Through its use began to show good results with the students’ performance, there was a major inconvenience: the existing tools do not allow professors to add new problems to the database so finding problems suitable to the multiple objective of the course became a hard task.

Given this problem and aware that the automatic evaluation supports to solve the above [2, 3], the development of computer system SEPAP (Spanish acronym of "Evaluation System for Learning Programming") started with the aim of helping students in reviewing their solutions. This tool has the functionality to verify the student’s solution with multiple test cases including extreme cases where the student fails often related to their lack of experience in some areas.
3. SEPAP project

In January 2011 we undertook the development of SEPAP in collaboration with a group of four Computer Science and Technology students. During this period, we started the requirement engineering process by defining the scope and primary goals of the project. In the first development stage, the tool provides options to add problems and test cases designed by professors. It also includes functionalities such as load groups, and above all, allow students to log in and upload solutions. The system was adapted to compile and accept solutions developed in Java or C/C++ and to provide feedback based on the solution submitted in case of acceptance or failure.

3.1 The functionality of the tool

SEPAP works with three kind of user account with different access privileges: student, teacher, and administrator. To enter the system prompts the user registration and password. Figure 1 shows the welcome screen.

SEPAP includes a list of problems that students can solve. The list of problems can be accessed in order to start working on a solution at the “Problem list” button. A screen of this list can be viewed in Figure 2.

To upload a solution, it is required the number of the problem, the language used for the solution, and an attachment of file with the code. Figure 3 shows the screen to submit a problem.

First part to use SEPAP it that the professor design a problem, this problem should have a clear description, an appropriate group of cases to be evaluated, this cases should have a input and output files. The Figure 4 shows the form where a problem could add to the tool.

A professor should add groups, add students. The Figure 5 shows how the professor could manage their groups.
Once the student select “submit solution”, SEPAP automatically provides feedback which can be ACCEPTED, COMPILATION ERROR, TIME LIMIT EXCEEDED or REJECTED. The professor could see the reports of the preformance of their students. The Figure 6 shows how SEPAP reports the preformance of a group.

SEPAP version 1.0 was completed in its development stage in May 2011.

3.2 Courses and students involved

Following this, finding the most appropriate course for the semester August-December 2011 was needed to tests our tool. The pilot program started with two groups of the Programming and Data Structures class (TC1005) as its students already have a technical background from the previous required course (TC1002 Programming Fundamentals). By the time the students take this course (sophomore year), they already know the basic programming concepts and are more experienced allowing them to take fully advantage of the evaluation system still under development phase. It was decided that the first programming language accepted by SEPAP would be Java, the same one used in the pilot groups.

In the first test, at the beginning of the session we found out that the students struggled understanding the process to be followed in order to upload and verify solutions using the SEPAP problems. Nevertheless, after overcoming the learning curve, the students began to create and submit their own solutions with no further delays. During the hour and a half session, the students worked on the development of the solution of two problems. At the end of the class, SEPAP had accepted the solution of more than 75% of the students. All this empowered students’ confidence, and after allowing late submissions during the rest of the day, a 90% of them got their solutions accepted into the system.

In the second and third test sessions, the students were already familiar with the platform and were encouraged to focus only on the development. In this way, the students were more attentive on working in their test cases before delivering a desktop solutions. After working on these two groups, three more groups were added to the program so the tool was also used by students enrolled in the Analysis and Design of Algorithm class (TC2001) in which the official development language is either C or C++. The students enrolled on this course are in their junior year and therefor have a stronger technical background and are more experience so they can work with more sophisticated algorithms. While the students taking the Data Structure class faced problems more basic problems usually related to data sorting, the ones taking the design of Algorithms class were working with search techniques such as backtracking (Knapsack problem), DFS and BFS algorithms. In order to increment the challenge, SEPAP allows the professor to add memory and time constrains to the solutions so students are force to work on not only correct solutions but also fast, elegant, feasible and applicable to real scenarios. With this new constrains added, the students had to avoid unnecessary workload in their solutions.

3.3 Data analysis

We gather information focused on two area of opportunity: the way SEPAP helps students on the learning process and about the design and usability of the system.

We designed a survey where all students of the 5 groups answered the following questions:

**SEPAP AS A TOOL FOR LEARNING PROGRAMMING**

- a. How many times have you used the system?
- b. Does the use of this tool supports to reinforce the topics seen in class?
- c. Do you think you could use SEPAP to increase your ability to develop programs?
- d. Is SEPAP easy to use?

**DESIGN PLATFORM**

- a. How would you evaluate the interface?
- b. How would you evaluate the steps to find a problem?
- c. How would you evaluate the steps to upload a solution?
- d. How would you evaluate the response time to a solution?
- e. How would you evaluate the history of your solutions?
- f. How do you rate SEPAP overall?
4. Results for SEPAP

The feedback provided by the students enriched the next versions and growth of SEPAP.

The data was obtained from 5 different tests from 3 different courses. The problems proposed go from a basic to an advance level involving solutions developed in Java, C and C++.

4.1 Results for learning programming

In the Figure 7 we can find the answer of the student to the question Does the use of this tool supports to reinforce the topics seen in class? And the analysis shows that more than 65% think is very helpful the use of an automatic tool additional to the sections in classes.

In the Figure 8 we can see that the tool is accepted perfectly by the students because to the question Do you think you could use SEPAP to increase your ability to develop programs? The are 100% agree with this tool to increase their abilities to develop programs.

In the Figure 9 we can discover that the tool is not so easy to use at the beginning because the results to the question Is the SEPAP easy to use? Shows that the student have difficulties to understand the use of a tool of this kind, because the results are so similar between agree and disagree.

We can discover that SEPAP is a very good tool to use in programming classes but we have to guide a little more at the first use of this may be with a problem during the class.

4.2 Results for design platform

We evaluate the design platform with five questions that tried to know the appreciation of the students about how they feel the use of the platform in the interface use.

In the figure 10 we can see that the 100% of the student is agree or almost agree about the physical interface, because the answered in this way to the question How do you evaluate the interface?
We can see that they take some minutes to find a problem, because the answered not so good to the question How would you evaluate the steps to find a problem? We can see the results of this question in the Figure 11.

Almost 90% thinks that is easy the way to upload their solution to a problem. In the Figures 12 we can see the results of this question.

One of the key aspect when you are using a authomatic tool for solve problems, is the timing that te tool takes to answer a sumbition, in this case all the students were very agree that SEPAP gives a fast answer of their sumbitions.

Fortunately the students thinks that SEPAP provides excellent answer time, and this could help to them to apreciate faster how their solution was as the response regarding expected.

We can see in the Figure 13 the positive results of the question How would you evaluate the response time to a solution?

Another important aspect of this kind of tool, is how it save our sumbitions, and almost the 80% of the students think that SEPAP handle very good the historial of their submitions. In the Figure 14 we can see the results of this question.

4.3 Summary of Results.

The results obtained from the end of course survey provided an excellent response from students to SEPAP:

- 82% of students believe that the tool supports the generation of strong skills to develop programs.
- 81% of students believe that the system is very user friendly and easy to use.
- 95% of students believe that the response time is excellent.

For the first test, it took 4.7 submissions for a solution in order to get accepted to the students. In contrast to the third test where it only took an
average of 1.4 submissions for their solution to get accepted.

These results reflect how students are gaining more awareness to send complete solutions for their activities, and we believe that SEPAP supports students in their learning of programming by providing a medium through which they can effectively develop their programming skills.

In the Figure 15 we can see the answer of the students to the question How do you rate SEPAP overall? Where more than 90% think that in general SEPAP is an excellent platform to support the topics seen in programming classes.

![Figure 15: How do you rate SEPAP overall?](image)

5. Conclusions and future work

The surveys reflected similar results regardless the fact that student students were enrolled in different courses with different programming languages and were solving problems of different nature and degree of complexity. They all agree that this type of tool can reinforce the course topics.

The results lead to the conclusion that a package of this nature will allow students to:

- Have a set of problems covering the study material in their multiple classes.
- Develop problems related to specific topics and increase (or diminish) the complexity by creating the correct test cases or setting up certain constraints.
- Obtain automatically a review of the solutions of the students.

The SEPAP is being widely spread within the institution and adopted by more professors willing to test the increased programming skills related to the use of automated evaluation tools. In addition, this provides more feedback and expands the range of reports to support the tool and the resolution of existing problems.

6. Acknowledgments

We thank Rafael García, Humberto Garza, Alan Gandarilla, and Roberto Plancarte for their valuable work on the development of the project.

We also thank Yolanda Martínez for being part of the professor who helps testing the tool in her groups and Juan A. Nolazco for his unconditional support for this project.

7. References