A PERSONALIZED COURSE GENERATION SYSTEM BASED ON TASK-CENTERED INSTRUCTION STRATEGY

Heba Elbeh and Susanne Biundo

Institute of Artificial Intelligence
University of Ulm,
D-89069 Ulm, Germany
<firstname>.<lastname>@uni-ulm.de

ABSTRACT
The Course Generation System (CGS) is a system that generates a course accommodate to the teaching material, student status and learning goal. In this paper we present a framework for authoring/course generation system using ontology and an HTN planning technique called PANDA.TUTOR. This system allows the author to prepare the course structure and content, enriched with classification information without need to define the adaptation rules or specify configurations for each student, aiming to keep all the authoring process low in terms of time and effort. The course generation system, which is basically an HTN planning system, introduces a personality based planning approach, in which the course structure and the course content are generated and adapted according to students’ personality and state. The system enriches with different learning scenarios, teaching strategies and learning styles. The student’s personality type taken as a guide to select an appropriate scenario, teaching strategy, and learning style in order to regulate the current emotional and motivational state of student.

1. INTRODUCTION
Intelligent Tutoring System (ITS) is computer based learning which assists students in their learning process. Thus, it has the ability to be adaptable according to the needs of students. During the learning process, it is important to individualize the given course and teach students according to their personality types, cognitive capabilities and current emotional and motivational states, to achieve the learning goal efficiently. Thus, students with various personality types differ in their behavior. The process of arranging personalized adaptations is usually complex. Consequently, the current platforms usually do not provide more than a relatively simple way of personalization and adaptation. Although most of course generation systems generate a suitable course according to the student’s cognitive ability, and gives some consideration to the motivation and emotion. The student’s personality type, emotional appraisal and coping ways and different regulation strategies for each personality type have not been considered so far.

Course generation has long been a research field which is also known as curriculum sequencing. Two main approaches are defined in course generation field [1]: the adaptive course generation and the dynamic course generation. In the first approach, the course is generated before it delivers to the student. Indeed, it could help the student to realize the complete view about the requested concept and provides free navigation ability through the presented part. In the second approach, the student progress is observed during the interaction with the presented course, then, the presented part is adapted according to the student goals and progress. Although, this approach is appropriate with the changeable state of student, it may cause confusion for the student when s/he moves from one part to another and may not get the complete idea. Thus, our system follows the first approach.

In this paper we will introduce our system PANDA.TUTOR, a new approach of course generation in ITS, that generates a personalized course as a structured representation of the subject based on Task-Centered Instruction Strategy. The objective of our system is to build a general Authoring/Course Generation system. Thus, the course content consists of a textual description, examples, exercise and the like is prepared by the author and enriched with classification information. The course generation system is an HTN planning system. It generates the course by tailoring the learning content to an individual learner, considering the student’s goal, the current emotional and motivational state of the student. Therefore, PANDA.TUTOR includes different learning strategies, learning styles and regulation strategies which help the planner to construct high individualized courses, and help students to build up their knowledge. In addition, it has the ability to adjust or maintain the student emotion.

The authoring part in our approach is defined over the course module and the course generation part is defined over the pedagogical module. Moreover, the aim of authoring phase is to
keep all the authoring process low in terms of time and effort. So, we will introduce an ontology that describes the course content module of our system. Thus, developing of the course ontology is a step toward creation shared and reusable adaptive educational systems. Also, the ontology permits the retrieval of learning materials after the pedagogical module generates the course. The proposed ontology model is a general educational ontology based on knowledge objects and the Instruction Design theory, which are introduced by Merrill [2]. Furthermore, we have considered a separation between the course Content reusability and the learning objects reusability. The authorized course afterwards is considered during the pedagogical module to construct a personalized lesson for the student.

Before introducing our proposed architecture in section 3, we will define the course generation, the relation between them and personality and learning styles in Section 2. In section 4, we will demonstrate the course ontology in our approach. Section 5 demonstrates how the HTN planning paradigm can be used to generate the personalized course. The paper ends with some concluding remarks in section 6.

2. ONTOLOGY AND E-LEARNING

The use of ontologies is a way of describing the semantics of information on the Web. Ontology provides a set of terms which should be shared among all the authors, and hence could be used as well-structured shared vocabulary. Ontology is a research domain helping us to overcome the most common problems in intelligent tutoring systems[3, 4]. It enables the ontology designer and the author to share a common course structure, and the educational materials to be reused. Besides, It allows us to specify formally and explicitly the concepts that appear in a concrete domain, their property and their relationships. Aroyo et al. [5] describe how an assistant layer uses an ontology to support the complete authoring. For the course structure, Henze et al.[6] propose an approach that describes the features of the domain ontology and the learner ontology, as well as observations about the learner’s interactions. Urllich[7] used the ontology to assemble the learning resources to generate a curriculum, taken into account the knowledge state of the student, the preferences and learning goals. On the other hand, there are standard meta-data, which are used to describe index, and search teaching materials; for instance, IEEE LOM, SCORM and IMS. However, these standards do not include any domain ontologies which can be specified, building on formalisms. In addition, Possible types of learning resources in LOM (Diagram, Figure, Table, Exercise, Text, Exam), mix instructional forms and resource type. On the other hand, many non-SCORM systems such as DCG [8], [9] support the pedagogical approaches but they lack the interoperability and flexibility, and they are do not support the reusability of educational materials. The contribution of our model is to build a general cognitive and constructive conceptual construction of the ontology that can be used in different types of courses. We consider a new approach to construct our ontology based on Instruction design theory of Merrill[2] as we will explain in the next sections.

3. LEARNING STYLES

Learning styles are defined as "the term learning styles is used to describe the attitudes and behavior that determine an individual’s preferred way of learning” [10]. Honey and Mumford itemize four learning styles as follows:

- **Active learning styles:** they prefer activity-oriented learning materials with high interactivity level, and become bored with repetition.

- **Reflective:** they like to deliberate on their experiences and study the situation from different perspectives, as well as collecting and analyzing the data before taking action. Those students prefer example-oriented learning material.

- **Theorists:** They strain to formulate their experiences in theoretical or logical form, and motive trying on ideas, theories, and experiments. They prefer exercise-oriented learning material.

- **Pragmatists:** they are eager to try out new materials, but concentrate on the concept that can help them to achieve their task, and prefer to explore and discover concepts by more abstract level. The theory-oriented learning materials are convenient for this type.

3.1. Personality and Learning Styles

The term Personality is defined as a permanent pattern of characteristics that discriminates between people in their feeling thought and behavior. In PANDA.TUTOR we considered the personality types of Vollrath [11]. He defines eight types of personality based on three types Extroversion(E), Neuroticism(N), and Conscientiousness(C). The Neuroticism and Extroversion are related to stress, while conscientiousness is related to the ability of coping. These types as follows; Spectator (low E, low N, low C), Insecure (low E, high N, low C), Sceptic (low E, low N, high C), Brooder (low E, high N, high C), Hedonist (high E, low N, low C), Impulsive (high E, high N, low C), Entrepreneur (high E, low N, high C), Complicated (high E, high N, high C).

There is no single learning style that is best for all students. Many studies attempt to investigate the relation between the personality type and learning styles. Furnham [12] links between the big five factor of personality (OCEAN) and learning styles. He investigates that the Extroverts are fairly consistent with Activists and pragmatists learning style, while the introverts are reflectors. The Neuroticism is more probably theorists and reflector, and Conscientiousness interacted.
positively with Activists style and negatively with theorist learning style.

4. SYSTEM ARCHITECTURE

As depicted graphically in Figure 1, our architecture consists of three main modules: course module, student module, and pedagogical module. An authoring phase is defined over the course module and course generation phase is defined over the pedagogical module. So we have developed general educational ontologies for the student module and course module that can be used for different domains. Ontologies help us to share, reuse and reason about information. During the pedagogical module two main process are defined; the dialog process and the course generation process based on planning technique that generates the course, regarding the course module and student module.

The generated plan will be a set of querying statements that query the course ontology to retrieve the learning objects to generate the course for students.

PANDA.TUTOR includes four types of scenarios; Lesson scenario, revision scenario, companion games scenario and test scenario. The term scenario means a discrete sequence of steps inside the learning process. In these scenarios the emotional and motivational state are considered for generating the course for the concept goal.

5. COURSE ONTOLOGY IN PANDA.TUTOR

Merrill[2] defines a set of knowledge objects to describe the subject matter content or knowledge to be taught. These knowledge objects are considered as a framework to organize the knowledge base of content resources. The components of Knowledge objects are not specific to a particular subject matter domain, the same knowledge object components can be used for representing a variety of domains (e.g. mathematics, science, humanities, technical skills, etc.). He describes the contents in a way that they could be manipulated in a computer system to automatically create instruction from the content. Thus, he defines five types of knowledge objects information_about, parts_of, kinds_of(subconcept(s)), how_to (Process), and what_happens(condition or principle).

For each knowledge object, two levels of information are defined; information_Level (generality) and a portrayal_Level. By considering these Information levels, four instructional strategies are defined; (Presentation (TELL) and Demonstration (SHOW)) for Information_Level, and Activation (RECALL) and Application (DO)) for portrayal level.

The course space in our approach has two main sub-spaces; the course structure as the course domain space and the course content as the media space. The domain space consists of the most important concepts of course domain with different cognitive levels and relationships between them. In our work we considered the concept map approach for developing our course ontology. Concept mapping is a technique for representing concepts and their hierarchical interrelationship as a graph, which nodes represent concepts and arcs represent relationships among them [13]. While, the media space is a semantic network that is named the resource network, which is used to represent learning objects of course content. Learning objects is a unit of content of digital resources that can be shared and reused to support teaching and learning process[14].

The proposed domain is a framework for instructional design of knowledge objects from the viewpoint of ontology and semantic web. The domain ontology has been developed under OWL language. We have developed general educational ontology that can be used for different courses. This ontology is considered during the course generation to generate an adapted and personalized course. The protege 4.1 framework has been selected to edit and construct the contents.

Consequently, in our ontology six main classes are defined as a hierarchical fashion. The course space, the course concepts, the knowledge objects, the instructional strategy components, instructional objects and learning objects. Indeed, for each course, a set of concepts should be defined. For each concept different types of knowledge objects could be defined, for each knowledge object (KO) different instruction components (IC) can be defined, different instruction objects can be specified for each instruction component according to the course and the preferences of the author. Finally for each instruction object different media or learning object can be determined. In our ontology we will consider different kinds of relations; relations among Concepts, relations between the concept and its knowledge objects, relation between knowledge objects and its instructional components, relations between Instructional component and instructional objects and relations between instructional component and its teaching material types.
These layers as depicted in figure 2 are as follows:

- **The Course layer**
  \[\text{Course} \equiv \text{Concept}_1 \sqcup \text{Concept}_2 \sqcup \cdots \sqcup \text{Concept}_n\].
  Course class represents any course of the learning environment. It is defined as the main class which represents the subjects being taught in an educational application. The class course consists of a set of concepts that define the course. For example, the planning system or Artificial Intelligence could be individuals of this class. The course class contain several properties that describe the names and a brief description of the course: CourseName, CourseDescription and the objective. Also, different object properties (relations) are defined, for instance HasObjective whereas hasConcept (isConceptIn is its inverse) and hasResource relations.

- **The concepts layer**
  \[\text{Concept} \subseteq \text{Course}, \text{and Concept} \subseteq \exists \text{has KO.KO}\].
  The course in our ontology considers the concept as a main building block. Course concepts refer to basic concepts concerning the course. The concept holds one unit of knowledge and explains different aspects of it with different types of teaching material. Thus, the concept can represent as a bigger or smaller course structure. The ontology for this model preserves the relationship between concepts as in the concept map. Thus, the author can specify various relations between concepts. In addition, the concept structure is built to represent the domain ontology that provides the structure of the course. There are different relations among concepts are considered; for instance (hasPrerequisite, isPrerequisiteOf), RelatedTo or SimilarTo, OppositeOf.

- **Knowledge objects layer**
  \[\text{KO} \equiv \{\text{Information}_About, \text{Part}_of, \text{Kind}_of, \text{How}_to, \text{What}_happen\}\].
  KO \subseteq \exists \text{has IC.IC}. For each KO we should define a set of IS such that IS \in (Presentation (Tell), Demonstration (Show), Activation (Recall) or Application (Do)). We represented the Knowledge objects class as an abstract class of the Information_about, Parts_of, Kind_of, How_to and What_happen subclasses. Thus, there are relations between the concept and its Knowledge objects; hasInformation_about, isInformation_about_for, hasPart, isPart_of, hasKind, isKind_of, hasHow_to, isHow_to_of, and hasWhat_happen, isWhat_happen_of.

- **Instructional component strategies layer**
  \[\text{IC} \equiv \{\text{Presentation(Tell)}, \text{Demonstration(Show)}, \text{Activation(Recall)} or \text{Application(Do)}\}\], and \(\text{IC} \subseteq \exists \text{has IO.IO}\). With each knowledge object four instructional components can be defined: Presentation (Tell), Demonstration (Show), Activation (Recall) or Application (Do). The relations between the knowledge objects and Instructional Components are as follows; \text{hasActivation}, \text{isActivation_for}, \text{hasApplication}, \text{isApplication_for}, \text{hasDemonstration}, \text{isDemonstration_for} and \text{hasPresentation}, \text{isPresentation_for}.

- **Instruction objects layer**
  \[\text{IO} \equiv \{\text{Definition, Example, Exercise, Theory}\}, \text{and} \ \text{IO} \subseteq \exists \text{has LO.LO}\]. For each IS we should define a set of instructional objects (IO). Different IO can be assigned (by the author) for each type of instructional components. For instance, Definition, Example, Exercise, Cased_Study, Assignment, Test, Experimental, List, Condition, Action, Theory, Fact, Principle, Proof, Analysis, Description, History, Menimonic, etc. For example the author can assign Definition, or Theory or both for the Presentation component, and Problem, Case Study or Exercise for Application part.

- **Learning objects layer**
  \(\text{LO=(LO)}\), LO is teaching materials with different media type where media type \in (text, audio, video, ..., etc). For each type of instructional object, different forms of learning object can be assigned. The type of learning object as media type is defined the presentation format. For example text, video or audio can be assigned for the definition of the concept.

Note that, in our educational system we considered different cognitive levels for the course parts, the quantity level and Quality_Level. The quantity_LEVEL is defined by the author for the course part, while quality_LEVEL is determine according to the student achievement level. Also, Different types and levels of Example and Exercise.

6. **AUTHORING PART**

Authoring of adaptive Content is one of the most important activities of the course-based adaptive tutoring system. Our
goal is to help the (a non-programmer) author to build highly
reusable software components that can be employed in a large
number of scenarios. Thus, we introduce an ITS authoring
system aim to help the teachers to configure different au-
thorizable courses with minimum effort. It is for the teacher to
author, construct or modify different parts of the course and
incorporate new learning materials without need for an inter-
vention from the system developers. The authoring model
allows the author to include different contents in the same
concept. Moreover, there is no need to create or modify the
teaching method or strategies for each student. The planning
system plays the role of the teacher in our architecture. This
is opposite to other systems in which the author use the au-
thoring tool to define the Domain Model, Student Model, and
rules for the Teaching Model. The author with the course
module has two major missions. The first, define the course
structure; it is the knowledge base of the course. The second,
defines the Repository by storing all the learning and test ma-

Algorithm 1: Authoring Algorithm

1. Step 1: Define the course objectives of the course.;
2. Step 2: According to the course learning objectives, add
courses accordingly, and finally get: course
   = \{Concept i | i = 1, 2, \ldots, m\};
3. Step 3: define different relations between concepts;
4. Step 4: Define the data property Concept of the concept;
5. Step 5: For each knowledge object of concept i =
   \{Knowledge object j | j = 1, 2, \ldots\}
   such that Knowledge object= (information about, Part_of,
   kind_of, how_to, what_happen);
7. if the concept not further has KO then
8.   Go to step 3 to define the next concept.;
9. else
10. Define the different instruction strategies(IS) for its KO.
   IS= Presentation, Demonstration, Activation, and
   Application;
11. foreach IS do
12.   Define a set of IO. IO= Definition, Example, Exercise, ..;
13. foreach IO do
14.   Connect it with its learning object;

7. HTN PLANNING FORMAL FRAMEWORK

Our AI planner relies on a domain-independent hybrid plan-
ning framework [15]. Hybrid planning [16, 17] integrates hi-
erarchical task network planning [18] with concepts of partial-
order-causal-link (POCL) planning. The resulting systems
integrate task decomposition with explicit causal reasoning.
Therefore, they are able to use predefined standard solutions
like in pure HTN planning. In our framework, a partial plan
P = \{TE, \prec, VC, CL\} consists of a set of plan steps TE, i.e.
(partially) instantiated task schemata, a set of ordering con-
straints \prec that impose a partial order on the TE, and a set
of variable constraints VC. CL is a set of causal links. A
causal link \{te_i, \varphi, te_j\} specifies that the precondition \varphi
of plan step te_j is an effect of plan step te_i, and is supported
this way. A domain model D = (T, M) consists of a set of
tasks as well as a set of decomposition methods. A task
t(t) = \{prec(t), add(t), del(t)\} specifies the precondi-
tions as well as the positive and negative effects of a task.
 Preconditions and effects are sets of literals and \tau = \tau_1, \ldots, \tau_n
are the task parameters. Both primitive and abstract tasks
show preconditions and effects.

A method m = (t, P) maps an abstract task t to a partial plan
P, which represents an (abstract) solution or “implementa-
tion” of the task. In general, each abstract task has imple-
mented by a number of different methods. As opposed to typ-
ical HTN-style planning, no application conditions are associ-
ated with the methods. A planning problem \Pi = (D, S_{init}, P_{init})
consists of a domain model D, an initial state S_{init}. P_{init} rep-
resents an initial partial plan.

Refinement steps include the decomposition of abstract tasks
by appropriate methods, the insertion of causal links to sup-
port open preconditions of plan steps and the insertion of plan
steps, ordering constraints, and variable constraints.

8. COURSE GENERATION IN PANDA.TUTOR

In PANDA.TUTOR we aim to develop an adaptive learning
environment in which the course content and pedagogical as-
pects are adapted for each student, considering the personality
type, educational progress, learning style, cognitive level
and the emotional and motivational states of the student. The
integration of cognitive, emotion and motivation with ITS as-
sists students to achieve their goal. The design of the adap-
tive learning system requires a huge number of rules which
represented in most systems as if - then rule. The proposed
methodology is based on an intelligent mechanism that tries
to mimic an instruction designer model. All the adaptation
rules are modeled in the planning domain.

According to the current state of the student module, the
planner attempts to find a sequence of operators which can
achieve the student goal. An operator is much like a rule in
a production system. Using the preconditions and expected
effects of operators, the Planner can simulate the invocation
of the various steps of a teaching plan, and thereby determine
whether the plan is likely to be successful. A successful plan
is one which can be applied to the current state of the student
module to achieve the given goal.

PADNA.TUTOR is a constructive course generation which
helps the student not only to complete their course more ac-
curately and efficiently, but also to build new knowledge in
enjoyable ways. Thus, we follow deep learning rather than
surface learning. The deep learning approach encourages stu-
dents to understand the concept, relate new content to the pre-
vious knowledge, which helps students to structure and or-
organize their knowledge, which is known as task-centered instruction strategy. This approach incorporates students in the whole task early in the instructional sequence. In which, each topic in a given area is taught in turn.

The instruction starts by demonstrating the first whole task in the progression. The first demonstration should be a complete task but it should be the least complex version of the whole task in the progression. This demonstration forms the objective for the task and provides the context for the students. The second task asks about the first task and introduces more information, and so on. That helps the students to easily grasp a demonstration of the whole task. Thus, getting students involved with realistic whole situations will help them to form appropriate schema and mental models. As a result, it facilitates doing application when students try to solve a new acquired knowledge and skill[2]. While the surface learning approach is interested only in a part of material and memory facts, ignoring the attempt to construct the student’s knowledge. So, instructional materials in our approach that are generated for a particular learning goal are organized around specific key concepts. Our plan is generated according to the course structure especially the concept goal structure and its relations with other concepts and the student module.

The educational material provided for each concept, is organized in different levels of performance which the student should achieve in order to master the concept. Instruction strategies are adapted for the presentation of the educational material follow the student learning style.

Adaptation of the course content implies that different students receive different course version of the same concept. Our approach is adapted according to: the scenario type, the level of depth (quantity level), the student level and goal (ranging from high level overview to in-depth explanation), the level of difficulty (quality level), the learning styles, the different motivational states and different emotional states. At the beginning, the student determines the required course and the system presents the learning content according to the student level. After that, the student selects a concept from the concept list of the course. Then, the type of learning scenario can be selected either by the student or by the system according to the student’s emotional, motivational state and the performance history in the selected concept. Then the system presents a course for the selected concept.

For instance, the abstract task of the lesson scenario is Generate_Lesson_Scenario, for variables ?st of type Student, ?pers of type Personality, and ?Cname of type Concept. The schemata of this task is as follows:

\[
\text{Generate_Lesson_Scenario(?st: Student, ?pers: Personality, ?Cname:Concept)}
\]

The intended semantics of this schema is that the student is to be taught the specified concept. On this level of abstraction neither the learning style nor emotional and motivational state are relevant, the only significant state change concern the student personality type. We have two refinements, in which the concept is studied for first time or studied before. For the first one the abstract task is decomposed into a task network with six sub-tasks, in which the emotion and motivational states are considered and then the generated course is manipulated. With the second refinement the abstract task is decomposed into three sub-tasks according to the performance history of the student’s cognitive level for the respective concept.

In the first method (the concept is studied for the first time), the following sub-tasks are defined Consider_emotional_State, Consider_Motivational_State, Teach_first_Lavel, Teach_Second_Lavel, Teach_Third_Lavel, Teach_Fourth_Lavel. The first two tasks are organized according to the emotion and motivation strategies that manipulate different emotional and motivational states. Different refinements are considered for Consider_emotional_State such as increase happiness, decrease sadness, decrease fear. The refinements of motivation task Consider_Motivational_State are used to increase confidence, increase effort, maintain confidence. The actual learn procedure, encoded by Teach_first_Lavel, Teach_Second_Lavel, Teach_Third_Lavel and Teach_Fourth_Lavel.

In our learn context, according to Task center instruction strategy, the learning process is organized in four levels:

1. In the first level, a presentation(Tell) (overview, definition, introduction,...) for the whole view of concept is introduced. This demonstration forms the objective of the concept. As well as presentation(Tell) and demonstration(Show) about the part(s) of the concept.

2. In the second level, we deliver activation (remember or recall) and application (Do) for both concept and the concept’s part(s) and then teach the student presentation and demonstration about the sub-classes of concept with different learning styles.

3. In the third level, the activation application is delivered for both concept part, sub-classes and Process(How to) of the selected concept. Also, presentation and demonstration about the principle (what happen) with different learning styles are delivered.

4. In the fourth level, the student is asked to apply allKO of the concept. Finally, deliver an assignment about the whole concept to assess the student's level.

Note that, the task Select_Learning_style connects between the personality type and the appropriate learning style. For instance, the appropriate learning style for the personality type
Entrepreneur (high E, low N, high C) is the Pragmatist style. For teaching each knowledge object, four different methods are defined to represent the learning styles. For instance, to model the Pragmatist style for the concept’s process (How to), the generated plan will call demonstration about the process, then presentation about the concept’s process, and finally application about the process. While, in the Reflector style, the presentation will be generated then demonstration and application. The connection between the personality type and the appropriate learning style will be considered during the domain of the planning. Note that, for each KO we considered different learning styles.

9. CONCLUSION

In this paper we introduced a new personalized authoring/course generation system PANDA.TUTOR. The goal of this system is to construct the student knowledge and help the author to represent the required course. Accordingly, a new approach is introduced for the course module using ontology technique, as well as for modeling the pedagogical module using the HTN planning.

ACKNOWLEDGEMENTS

This work is done within the Transregional Collaborative Research Centre SPB/TRR 62 “Companion-Technology for Cognitive Technical Systems” funded by the German Research Foundation (DFG).

10. REFERENCES


