Adaptive Patterns for Intelligent Tutoring Systems

Dina Salah, The American University in Cairo, Computer Science Department, dsalah@aucegypt.edu
Amir Zeid, The American University of Kuwait, Division of Sciences and Engineering, azeid@auk.edu.kw

Abstract

The complexity of design and implementation of Intelligent Tutoring Systems (ITS) is caused by the lack of a clear road map or implementation methodology. This has led us to investigate the role of patterns in ITS implementation in order to provide software developers with solutions to recurring ITS design problems.

In this research work, we highlight the role of adaptive patterns in intelligent tutoring system implementation. We explain how those patterns were used in building The Arabic tutor, an Intelligent Language Tutoring System over the World Wide Web for teaching a subset of the Arabic Language that combines the flexibility and intelligence of Intelligent Tutoring Systems with the availability of the World Wide Web applications.

The implementation process of the Arabic Tutor was our proof of concept for the validity and usefulness of adaptive patterns in ITS implementation.

1. Introduction

ITSs incorporate built-in expert systems in order to monitor the performance of a learner and to personalize instruction on the basis of adaptation to the learner's learning style, current knowledge level, and appropriate teaching strategies (Liegle & Woo, 2000). The classical ITS architecture is composed of the following components (El-Sheikh & Sticklen, 1998):

- Expert Model: This model contains the domain knowledge.
- Pedagogical (Tutor) Model: This model provides the knowledge infrastructure necessary to tailor the presentation of the teaching material according to the student model.
- Domain Model: This model contains the knowledge about the actual teaching material.
- Student Model: This model stores details about the student's current problem-solving state and long-term knowledge progress, essential for adapting the material to the student's characteristics.
- Communication (User Interface) Model: This model is responsible of user interaction.

A recent analysis of a number of existing ITS architectures has revealed that many ITS designers and developers use their own solutions when faced with design problems that are common to different systems, models, and paradigms. However, a closer look into such solutions and their comparison often shows that different solutions and the contexts in which they apply also have much in common, just like the corresponding problems do. In all such cases we can talk of the existence of patterns (Vladan Devedzic & Harrer, 2004).

Adaptive instruction can be defined as real time modification of the instructional curriculum, learning environment to suit different student characteristics ("ELEN

Project”). Adaptive patterns are patterns that are used in Adaptive Learning Systems. However, we found that there is a number of adaptive learning patterns that can be of great use to the designers and developers of ITSs if they were modified to suit the ITS developers and designers needs.

It is important to note that some of the discovered patterns were based on existing patterns; however, we modified some of these patterns in order to suit the needs of ITS implementation. Any modified pattern has a name that begins with the word “New”.

2. Adaptive Patterns for Building the Student Model in Intelligent Tutoring Systems

The process of building a typical ITS is composed of four phases; building the Domain Model, Student Model, Tutor Model and the Graphical User Interface Model. However, this is the typical sequence followed, nevertheless, it is not mandatory to strictly follow this order and some ITS developers use a different order for building ITSs.

In the following section we will highlight the adaptive patterns that can be useful in building the student model.

2.1 Pattern 1: User Model Definition

Classification: Adaptive Patterns
Intent: An ITS enriches its functionality by maintaining a student model and providing mechanisms to modify application features based on that. These modifications can be deciding to move to a higher difficulty level, showing the student more detailed feedback, etc., thus resulting in a personalized instruction. Standardization of the student model is an important issue, because through it we can greatly enhance the student model’s portability. This will allow learners to use several different ITS(s) and to carry their personal model with them, providing the systems with the same image of themselves, without that leading to compatibility problems. We need a student model that is small, compact and flexible. What information should an ITS keep for the student in order to offer him the best possible learning experience?
ITS Applicability: In an ITS setting the items that should be included in the user model definition have to be directly related to the user as a learner – anything that would be considered useful to better adapt to the learner’s particular characteristics. A complete user model definition should be comprised of the following elements:
1. **Demographic data**, which are relevant to the particular ITS (e.g. as age, gender, etc.)
2. **Student goals**, which are related to the long term and short term learning goals.
3. **Student preferences**, which includes the mode of delivery, accessibility requirements, or assessment.
4. **Student knowledge**, which includes the knowledge level about concepts to be learned and weaknesses and strengths on particular areas, sections or points of the concepts.
5. **Usage data**, which includes information like which pages were viewed, in what order, etc.

6. The **Stereotype** that applies to the student, which essentially is the group of learners he belongs to based on some predefined presuppositions in terms of knowledge level, learning and cognitive styles.

Known Uses

1. **An Interactive Course Support for Greek** (Tudor Heift, Toole, McFetridge, Popwich, & Tsiplakou, 2000): An ITS that teaches the Greek language.
2. **The Web Based German Tutor** (Trude Heift, 1998): A web based ITS for teaching German.
3. **Web Passive Voice Tutor** (Web PVT) (Virvou & Tsiriga, 2001): An ITS for teaching English passive voice.

Both of (Tudor Heift, Toole, McFetridge, Popwich, & Tsiplakou, 2000)and (Trude Heift, 1998) has three student stereotypes beginner, intermediate and advanced. Feedback messages are customized to suit current student knowledge level where beginners receive more detailed feedback than advanced students. Whereas (Virvou & Tsiriga, 2001) has four student stereotypes novice, beginner, intermediate and expert. Nevertheless, all the above mentioned ITSs take into consideration the student stereotype, knowledge, short term goals and usage data, but none of them take into consideration student demographic data, long term goals or preferences.

Related Patterns: User Goals, New Student Model Initialization

References: The User Model Definition Pattern is one of the patterns that were used in adaptive systems ("ELEN Project"); however, we discovered that this pattern can be useful to ITS developers and designers.

2.2 Pattern 2: User Goals

Classification: Adaptive Patterns

Intent: The student model description is a part of the student model component of an ITS and it should include student goals. What information should be considered as student goals in a student model in an ITS?

ITS Applicability: Any ITS needs to include specific student goals in the student model in order to facilitate adaptation, and capture the real intent of the learner with respect to the learning material.

Student goals can be divided into two categories:

Long-term goals: Educational goals that are valid for a longer period of time and require significant effort to be met, these goals are usually determined by the learners.

Short-term goals: Educational goals that are valid for a shorter period of time and require relatively moderate effort to be met. These goals are usually determined by the Tutor Module component of an ITS.

Known uses

1. **An Interactive Course Support for Greek** (Tudor Heift, Toole, McFetridge, Popwich, &Tsiplakou, 2000): An ITS that teaches the Greek language to foreigners
2. **The Web Based German Tutor** (Trude Heift, 1998): A web based ITS for teaching German.
3. **Web Passive Voice Tutor** (Web PVT) (Virvou & Tsiriga, 2001): An ITS for teaching English passive voice.

All the ITSs mentioned above accomplish short term goals by tailoring teaching material content and sequence to suit current student knowledge level.

Related patterns: User Model Definition, New Student Model Initialization.
References: The User Goals Pattern is one of the patterns that were used in Adaptive systems ("ELEN Project"); however, we discovered that this pattern can be useful to ITS developers and designers.

2.3 Pattern 3: New Student Model Initialization

Classification: Adaptive Patterns
Intent: ITSs initialize the student model before all interactions. What is the minimum amount of information needed to start the system? What kind of information and what amount is the student willing to provide?
ITS Applicability: ITS developers and designers need to initialize the student model with two types of knowledge; knowledge that is acquired from the students and knowledge that can be acquired automatically from the system.

The User Model Initialization Pattern initializes the student model with the student stereotype through one of the following methods:

- **User driven.** The user specifies explicitly what stereotype he belongs to.
- **Inferred by rules.** These rules indicate which user model elements and values can activate a stereotype.
- **Speculated by rules .**If the user does not specify his knowledge level then it is assumed to be average.

Known Uses
1. **Web PVT** (Virvou & Tsiriga, 2001): The student is initially assigned to one of the four distinct stereotypes, namely novice, beginner, intermediate and expert, according to her/his performance on a preliminary test.
2. **Verb Expert** (Fum, Giangrandi, & Tasso, 1989): At the beginning of each session, the tutor starts the interaction with the student by presenting him by an exercise on a given topic. The Student Modeller compares the answer of the student with that of the expert Module in order to identify the errors and to formulate some hypotheses about their causes in order to initialize the student model.
Related Patterns: User Model Definition, New Student Model Maintenance, User Goals.
References: This pattern was based on the User Model Initialization Pattern that is used in Adaptive systems ("ELEN Project"), however, we discovered that this pattern can be modified to suit ITS developers and designers needs.

We modified the solution suggested by the User Model Initialization Pattern in the New Student Model Initialization Pattern by accurately initializing the student model using an exam generator component. More details can be found in section 3.3.

2.4 Pattern 4: New Student Model Maintenance

Classification: Adaptive Patterns
Intent: During the course of interaction with ITSs many things about the student can be changed, e.g. current student knowledge level. Thus, the student model must be adapted to the new realities. How should the system capture those changes so as to maintain a good student model ("ELEN Project")?
ITS Applicability: ITS designers should define the conditions that govern the maintenance of the student model. The designer should define the scope of the

maintenance changes by defining the reason for updates. The reason is then quantified in terms of choice of elements to undergo change.
The Student Model Maintenance Pattern suggests maintaining an accurate student model through one of the following methods:

- Using a questionnaire form that indicates the amount of benefit the user gained from using the system.
- Interactive update of the user model by showing a pop-up form requesting the user to answer a question.

We modified the solution suggested by the Student Model Maintenance Pattern by using an exam generator component to conduct a per topic exam. This can occur after the student is presented by the topic learning material. The exam generator component can use the pool of questions supplied earlier by the subject instructor to randomly select a number of questions that can test the current student knowledge level and according to the results of this exam the user model maintenance module can update the student model to reflect his current status and the topics covered and his knowledge level per topic.

Known uses

1. **Web PVT** (Virvou & Tsiriga, 2001): A web based ITS for teaching the English passive voice. It records information about which concepts the student has mastered and to what extent. In addition, it records the kinds of error the student has made during past interactions as well as the most suitable explanation of each category of error. The information from the long term student model forms an individual model of the student, which together with the active stereotype are used in order to provide adaptive navigation support and perform intelligent analysis of the student's solutions to exercises.
2. **An Interactive Course Support for Greek** (Tudor Heift, Toole, McFetridge, Popwich, & Tsiplakou, 2000): A web based ITS for teaching Greek. The student model is a representation of the current skill level of the student. For each student the student model keeps score across a number of error types, or nodes, for example, grammar or vocabulary. The score for each node increases and decreases depending on the grammar’s analysis of the student’s performance. The amount by which the score of each node is adjusted is specified in a master file and may be weighted to reflect different pedagogical purposes.

Related Patterns: New Student Model Initialization.

References: This pattern was based on the User Model Maintenance Pattern that is used in Adaptive systems ("ELEN Project"), however, we discovered that this pattern can be modified to suit ITS developers and designers needs.

3. Using Adaptive Patterns in Building the Arabic Tutor

3.1 Pattern 1: User Model Definition

A student model is the image that the system has about the learner. The closer it is to the learner’s real characteristics and needs, the better the personalization. The User Model Definition Pattern was used in the Arabic Tutor; the user model included short term and long term student goals, student knowledge level on particular topics or learn items, usage data, and student stereotype.

3.2 Pattern 2: User Goals

User Goals Pattern was used in handling both long term and short term goals in the Arabic Tutor.

Long Term Goal Handling in the Arabic Tutor:

Students who use the Arabic Tutor are required to determine their goal from using the system. This is achieved by allowing the students to choose the topics that are of interest to them. These topics may be dependent on each other. Each topic may have successors, predecessors, or topics that can be taught in parallel with this topic.

Short Term Goal Handling in the Arabic Tutor:

Short term goals are goals that are valid for a shorter period of time, for example, if the long term goal of a certain student was "to learn topic X", then the short term goal is the presentation of the appropriate learn items and examples that he needs to study or view in order to gain the required knowledge to master the chosen topic. Short term goals in the Arabic Tutor are usually determined by the Tutor Model component.

The short term goal handling in the Arabic Tutor begins after the student signs up and determines his long term goals from using the system. The results of the entry exam per topic will be used to control the teaching material that will be viewed by the student .This is illustrated in table (1):

Table (1): Teaching Material According to Student Level.

Student Stereotype	Topic Explanatory Text	Learning Item Explanatory Text	Number of Learn Item Examples
Beginner	Yes	Yes	6
Intermediate	No	Yes	4
Advanced	No	Yes	2
Expert	No	No	1

In this phase, the student needs to perform a complex task consisting of several subtasks where decisions that need to be made in each subtask may not be known to the student. This complex task is represented by the student long term goal which is mastering a certain topic. This long term goal can be divided into a number of short term goals. Thus the complex task can be divided into a number of a subtasks represented in the learn items that are included inside this topic and that needs to be studied in a particular order to preserve the predecessor and successor relationship between learn items.

3.3 Pattern 3: New Student Model Initialization

The New Student Model Initialization Pattern was used in the Arabic Tutor in initializing the student model with two types of knowledge:

- **Knowledge that is acquired from the students:** This knowledge was determined with the guidance of both the User Model Definition Pattern and the User Goals Pattern.
- **Knowledge that can be acquired automatically from the system:** This knowledge was acquired by implementing an entry exam that uses an Exam Generator Component that randomly choose a set of questions that represent different difficulty levels thus can accurately measure current student knowledge level per topic (stereotype) instead of speculation or inferring as suggested in the User Model Initialization Pattern.

3.4 Pattern 4: New Student Model Maintenance

The New Student Model Maintenance Pattern was used in the Arabic Tutor by implementing an exam generator component to conduct a per topic exam. This occurred after the student is presented by the topic learning material. The exam generator component used the pool of questions supplied by the subject instructor earlier to randomly select a number of questions that can test the current student knowledge level and according to the results of this exam the user model maintenance module updated the student model to reflect his current status and the topics covered and his knowledge level per topic. This exam consists of 12 questions that vary in the difficulty level according to the current mastery level of student per topic. In addition, according to the level of the student that was determined in the entry exam he will be presented by a tailored feedback message in case he answered the question incorrectly.

Table (2) highlights the per exam configurable parameters in the Arabic Tutor.

Table (2): The Per Topic Exam Configurable Parameters in the Arabic Tutor

Student Level	Number of Exercise per difficulty level			Topic Explanatory Text	Learn Item Explanatory Text	Number of Remedial examples
	Difficulty Level 1	Difficulty Level 2	Difficulty Level 3			
Beginner	12	0	0	Yes	Yes	3
Intermediate	0	12	0	No	Yes	2
Advanced	0	6	6	No	Yes	1
Expert	0	0	12	No	No	0

4. Conclusion

In this research work, we showed the role of adaptive patterns in implementing the user model in ITS, we used the example of the Arabic Tutor, a web based intelligent language tutoring system for teaching a subset of the Arabic Language as our proof of concept for the validity and usefulness of adaptive patterns in ITS implementation.

We aim to provide designers of new or existing ITSs with a road map for user model implementation within the ITS domain.

However, one cannot talk of patterns in the ITS domain only in the context of adaptive patterns. On the contrary, there are many other kinds of patterns that can be helpful in ITS implementation; design patterns, access patterns, instructional patterns, and interaction patterns. Moreover, ITS complexity can be overcome by creating a pattern language for ITSs that can help software developers resolve recurring problems encountered throughout all of software development process of any ITS. In this way, designers of new or existing ITSs, especially inexperienced designers, can take advantage of previous design expertise and save precious time and resources.

In (Salah & Zeid,2009) we proposed PLITS, a Pattern Language for Intelligent Tutoring Systems that utilizes design, access, instructional, adaptive and interaction patterns in ITS implementation. PLITS includes rules and guidelines which explain how and when to apply its patterns to solve a problem.

References

1. Devedzic, V., & Harrer, A. (2004). Common Patterns in ITS Architectures. Künstliche Intelligenz, 18(3), 17-21.
2. El-Sheikh, E., & Sticklen, J. (1998). A Framework for Developing Intelligent Tutoring Systems Incorporating Reusability. In Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems: Methodology and Tools in Knowledge-Based Systems. Springer-Verlag. Retrieved.
3. ELEN Project. Retrieved May, 2004, from http://www2.tisip.no/E-LEN/outcomes.php
4. Fum, D., et al. (1989). Tense generation in an Intelligent Tutor for Foreign Language Teaching: Some Issues in the Design of the Verb Expert. In Proceedings of the Fourth Conference on European Chapter of the Association for Computational Linguistics. Manchester, England. Association for Computational Linguistics. Retrieved.
5. Heift, T. (1998). An Interactive Intelligent Tutor over the Internet, In Proceedings of the Proceedings of ED-MEDIA 1998, World Conference on Educational Multimedia, Hypermedia & Telecommunications, Charlottesville, VA. Association for the Advancement of Computing in Education (AACE)
6. Heift, T., et al. (2000). An Interactive Course Support System for Greek.Bourdeau,J.&Heller,R.(eds). In Proceedings of the ED-MEDIA 00,World Conference on Educational Multimedia,Hypermedia &Telecommunications, Montreal Canada. Association for the Advancement of Computing in Education (AACE)

7. Liegle, J., & Woo, H.-G. (2000). Developing Adaptive Intelligent Tutoring Systems:A General Framework and Its Implementations. In Proceedings of the ISECON Philadelphia, PA, USA
8. Salah,D.,Zeid,A (2009). A Pattern Language for Intelligent Tutoring Systems. In Proceedings of the EuroPLOP 2009.Germany.
9. Virvou, M., & Tsiriga, V. (2001). Web Passive Voice Tutor: An Intelligent Computer Assisted Language Learning System over the WWW. In Proceedings of the Proceedings of the IEEE International Conference on Advanced Learning Technologies. IEEE Computer Society. Retrieved.