

A Gradually Developing Adaptive Tutoring System as the Course Progresses

Khulood Gaid¹ and Eshaa Alkhalifa¹⁺

¹ Royal University for Women, Bahrain.

Abstract. Adaptive Educational Systems are able to alter the online course as per the needs of each student based upon online tests. These systems require a lot of time and effort required to design and build such courses. This paper offers a practical adaptive system that is automatically built as instructors upload their adaptive lessons and tests to the online site. The system asks the instructor to associate multiple choice questions that are incorrect with error pattern names and to associate the error patterns with lessons students need to review. The result is that the adaptable system is dynamically built as the course progresses. An instructor can use reports to extract common error cooccurrences and infer information about misconceptions at the end of the course.

Keywords: Adaptive learning, intelligent systems, student modeling, learning management system.

1. Introduction

Adaptive Systems are defined by Boticario et. Al. (2005) as systems that are capable of four main tasks;

1. To manage explicitly defined learning routes adapted to each user.
2. To monitor and record the activities of students.
3. To automatically infer user trends and store them into a learner profile system.
4. To act upon available knowledge by making recommendations, or altering the course of instruction

Brusilovsky (2004) indicated that adaptive educational systems went through three different generations. The systems that were designed during the first generation offered adaptive navigational support, adaptive presentation, direct guidance, sorting of links, hiding of links, and annotating links in different colors. The systems that were designed during the second generation focused on creating adaptive webbased educational systems with only a few adaptive features. They also focused on producing new techniques for adaptation as in AHA (De. Bra & Calvi, 1998) explored several approaches to link removal, while INSPIRE explored learning styles (Papanikolaou et al., 2003). These systems also focused on development frameworks to develop adaptable learning systems.

The third generation did not register any increase in the implementation of adaptive learning system in real courses except by those who designed their own systems. Instead, educators went on to use online learning management systems as in Edu20.org and Blackboard (Blackboard Inc. 2002). These systems allow educators to upload notes offer online quizzes, and grade-books.

Brusilovsky (2004) indicates that every single function that an LMS has, there is a matching function in adaptive systems that can do the task better. Yet, the influence of adaptive systems on education is extremely limited. An explanation to the cause of this issue is offered by Boticario et. al. (2005). They indicate that building adaptive scenarios is a very complex task that covers the full life cycle of the learning process. In

⁺ Corresponding author: Eshaa Alkhalifa. Tel.: +973 17764486
Email address: eshaa@ieee.org.

fact, the task is so daunting that it may dissuade many educators from following this path to build learning systems because after all that effort, the learning materials on the system may be outofdate.

SIETTE (Guzman et. at., 2005) is an adaptive testing environment that allows instructors to add tests, however, it does not allow the errors to be linked to complete lessons, such that the full course is covered by the student. The only existing platform to develop full adaptive hypermedia courses is AHA (De. Bra, et. al., 1998). AHA is an open source platform that can be downloaded for development of online hypertext adaptive systems. It is an extremely rich platform that allows possibilities to adapt the page content in different ways, presentation of links, with or without icons ahead of the links, to enable the same link to lead to different pages based upon the values in the user model. However, AHA requires a full course analysis and design prior to building the adaptable course. Although the overhead required to design such systems has been significantly reduced, it is still a daunting overhead to instructors who do not know about adaptable system architecture and how it works.

2. A Gradually Developing Adaptive Tutoring System

The system presented here builds the adaptation as the course progresses. While a course x is running at University y, an instructor can decide to use this system. He or she started by adding a few lessons to the systems in one of the common formats. The instructor then adds error patterns that are expected from prior experience and then tests that students have to pass to confirm that they have mastered the course. These tests have to be multiple choice questions at this stage of this project.

An instructor associates the questions in the test with the lessons, so if a student exhibits a specific error pattern, it will add that to the student profile and advise students on what lesson to go to in order to remedy the situation. The basic idea is that chapters are usually relatively distinct entities so the error patterns are either dependant on that chapter in particular, or in ones that preceded it in order. Since the system remembers error patterns previously entered, then both these situations are covered.

As it starts the error patterns have to associations between them based upon Bayesian or probability based analysis because they are formed independently. This does not mean that those associations may not be added later either though system learning, or through instructor experience.

This paper will start by presenting the Instructor's experience while using this system because it is the main focus of the work done here. It will then briefly explain a student's experience while using the system and then conclude.

3. Instructor experience while using the system

3.1. The Registration Step

- Every instructor must provide a valid username, password, email, security question and the answer.
- The instructor needs to remember the answer to the security question because it is the only way the password can be changed.
- The instructor will be sent an email from the administrator that contains a randomly generated code. This code is needed to identify the instructor.
- In this step the instructor will need to fill the fields with valid personal data such as the first name, last name, email to communicate by, phone number and profession. The terms and conditions agreement must be read carefully before ticking the check box for agreeing on it.
- In this step the instructor needs to register his class by adding the code he received by email from the site administrator who acknowledged this person as an instructor for this course. This code is going to be used by students in order to register for the courses offered by the instructor.

3.2. Defining A New Attribute:

- In this page the instructor can define classifications and general attributes that are added to the database. New course types can be added, e.g. Art, Business. New student levels can be added to the existing list (beginner, intermediate and advanced) as in levels associate with book parts, part1, part2, part3. Time limits can be associated with quizzes and exams.

- Error patterns can be added by instructors as per the needs of that particular course or chapter.
- The instructor is allowed to define new course types, prerequisites, levels, time associations and level types.
- Examples of current error patterns include the following; Inaccurate procedure (algorithm), Application error, Knowledge based error .

3.3. Defining New Courses:

The instructor need to enter a course name, choose a course type, course prerequisite and level. After finishing this step the instructor will be redirected to a page where he can add lessons. Also, objectives of the course are chosen allowed is up to four objectives. Then, the instructor is redirected to the questions page to fill up questions and other details.

3.4. Defining Questions For Tests:

- The instructor chooses the course name from a drop down list. Then the testing material question must be entered. This step is needed so the system can recognize where it can generate questions. The question must be entered in multiple choice format. More question formats will be added as this project is expanded.
- The question level must be entered then (beginner, intermediate and advanced). The level is used to associate with question levels with student levels in the course to assess students at their own level and to inform students of their progress level in the course.
- Then the learning objective that the question is satisfying must be determined. Objectives are defined when a new course is defined, and student profiles show what objectives the student has achieved so far.
- The instructor needs to enter possible student choices. The correct answer will receive a “none” in the error pattern slot, while all other answers can be associated with one error pattern. If an instructor does not wish to associate an error with an error pattern the word “none” or “inapplicable” can be used to leave the choice without associating it.
- If the instructor wishes to add another question he can click on submit. Or he can choose done if he is done with adding questions.

3.5. Viewing Lessons

This is done simply by selecting the course code then the lesson that is to be viewed.

3.6. Generating Student Reports

If the user chooses to go to this page, he will be redirected to an initial page where a list of tips is provided. All tests are similar to the entry test described above except that questions are based upon the lesson. Materials are only marked as done if a student passes the test.

4. Student experience while using the system

A student can perform five main functions with this system;

1. Registration and this is similar to an instructor’s registration with one exception in that an instructor can invite a student to register. A student can also add a course without an invitation to study with the person who set up the course.
2. Taking the entry test: this is a multiple choice question test which shows the student level in the topics that follow. It is taken only once and it is used to set the intial set of lessons a student has to take following that day. An instructor chooses the questions that qualify for the entry test and there is length limit to the number of questions.
3. Student Profile: Students can see their profile which is shown in Figure 1. It gives them a list of error patterns and lessons that need to be completed.

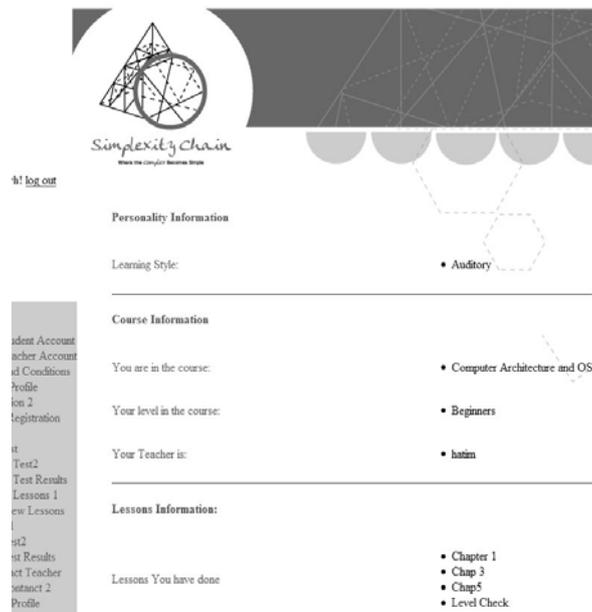


Fig.1: Student Profile

4. Viewing a lesson: A student selects the course and the lesson to see it. The lesson can be in any of the formats (ppt. doc. Images, video, etc.).
5. Taking a test; If the user chooses to go to this page, he will be redirected to an initial page where a list of tips are given. All tests are similar the entry test described above except that questions are based upon the lesson. Materials are only marked as done if a student passes the test.

5. System's Architecture

The system's architecture is shown in Figure 2.

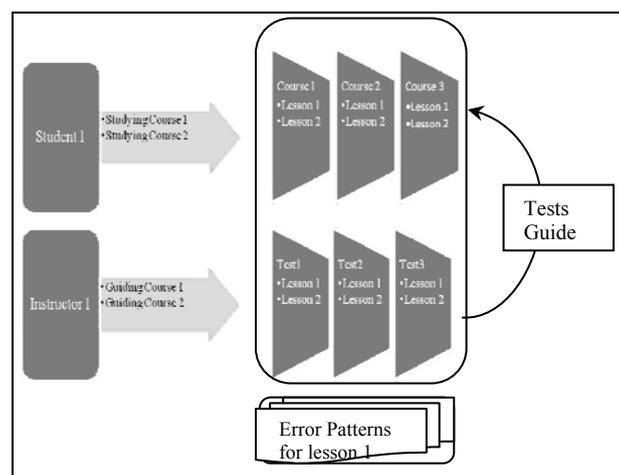


Fig.2 : Systems architecture

This system does not require a major course design effort prior to deployment. It only requires an instructor to do what they would normally do by uploading lessons in several different formats and uploading multiple choice questions in a way that is hardly any different from existing learning management systems.

The only difference is that for every wrong answer, the instructor writes an error pattern description in normal English and then assists students in recognizing what these error patterns are. This can happen while a course is running as and when a new test is uploaded.

As students progress through the course, they will see the lessons they have to take reduce in number as they pass their tests. An increase will occur if they demonstrate an error linked to a lesson that they did not

cover sufficiently well. In other words, the adaptive course is deployed when the first lesson is added. The database is designed to allow maximum flexibility for future expansion of the system.

6. Conclusion

The presented system has been tested on a test bed and will soon be deployed on the Internet. The goal is to make it available for use by instructors at the University and eventually to all those interested in using it in the world.

Future work possibilities are many. It is possible to work on altering the presentation of multiple choice question and to allow freeform answers to be submitted. It is also possible to add learning style preferences and to link that with lesson approaches taken or offered. The current system can accommodate different types of lesson materials including PowerPoint slides, Documents, Images, etc. A future system can select and recommend to students which would accommodate their style of learning.

Another possibility is to expand the error mining module. In its present form, it enables instructors to know if the materials may lack a bit of information that causes most students to display the same error pattern. This would enable instructors to modify their lessons to ensure that no error patterns shows behavior beyond the statistically normal behavior that is expected of it.

A central issue that may be raised by those versed in adaptable learning, is that they may say that each question in this system is an island. It is analyzed alone, and student profiles do not reflect any continuity of errors or styles.

That is true, for the way the system stands and works because it is the cost that has to be paid in order to be able to offer dynamically adaptable learning. However, it does not cause a problem to educators because the error analysis module that has been recently added assesses the co-occurrence frequency of errors to identify how they relate to each other. This will be an interesting module to review when the system goes live online to highlight to instructors relations in-between errors.

7. Acknowledgements

Both authors would like to thank god for blessing this effort.

8. References

- [1] J.G.Boticario, O.C. Santos, & P. Rosmalen. *Technological and management issues in providing adaptive education in distance learning universities*. EADTU, 2005. Retrieved Nov 14, 2010 from <http://www.eadtu.nl/proceedings/2005/presentations/plenary/Jesus%20Boticario.pdf>,
- [2] P. Brusilovsky. Adaptive Educational Hypermedia: From generation to generation (Invited talk). *Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education*, Athens, Greece, September 29 - October 3, 2004, pp.1933. <http://www2.sis.pitt.edu/~peterb/papers/PEG01.html> Accessed nov 14 2010.
- [3] P. De Bra and L. Calvi. Towards a Generic Adaptive Hypermedia System. *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia*, pp. 511, 1998.
- [4] Guzman, E., Conejo, R., and GarciaHervas, E., (2005). *An Authoring Environment for Adaptive Testing*, *Educational Technology & Society*, 8(3), 6676.
- [5] J.M. Nunes, S.P. Fowell. *Hypermedia as an experimental learning tool: a theoretical model*, *Information Research News*, 6(4), 1996, pp. 1527, <http://informationr.net/ir/21/paper12.html> Accessed 14 nov 2010.
- [6] K.A. Papanikolaou, M. Grigoriadou, H. Kornilakis, G.D. Magoulas: *Personalizing the Interaction in a Webbased Educational Hypermedia System: the case of INSPIRE*. *User Model. UserAdapt. Interact.* 13(3): 213267 (2003)
- [7] Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective HumanComputer Interaction: Second Edition*. AddisonWesley, Reading, MA. A description is available.
- [8] Blackboard Inc. (2002) Blackboard Course Management System 5.1, Blackboard Inc. <http://www.blackboard.com/> (Accessed 14 November, 2010)
- [9] Edu20.org (2006) Founded by Graham Glass. <http://www.edu20.org>