Surjono, Herman, *The design and implementation of an adaptive e-learning system*,
The International Symposium Open, Distance, and E-learning (ISODEL 2007),
Denpasar, Indonesia, 13-15 November 2007

The design and implementation of an adaptive e-learning system

Herman Dwi Surjono, Ph.D.
hermansurjono@uny.ac.id
College of Engineering, Yogyakarta State University

Abstract

This paper describes the design and implementation of an adaptive e-learning system that provides a template for different learning materials as well as a student model that incorporates five distinct student characteristics as an aid to learning: primary characteristics are prior knowledge, learning style and the presence or absence of animated multimedia aids (multimedia mode); secondary characteristics include page background preference and link colour preference. The use of multimedia artefacts as a student characteristic has not previously been implemented or evaluated.

The system development consists of a requirements analysis, design and implementation. The design models including use case diagrams, conceptual design, sequence diagrams, navigation design and presentation design are expressed using Unified Modelling Language (UML). The adaptive e-learning system was developed in a template implemented using Java Servlets, XHTML, XML, JavaScript and HTML. The template is a domain-independent adaptive e-learning system that has functions of both adaptivity and adaptability.

I. Introduction

There are many web-based learning and e-learning systems available on the Internet, but they provide only the same plain hypertext pages to all students regardless of individual ability. In many current web-based courses, the course material is still implicitly oriented for a traditional on-campus audience consisting of homogeneous, well prepared and well motivated students. However, web-based courses are used by a much wider variety of students than any campus-based courses. These learners may have very different goals, backgrounds, knowledge levels and learning capabilities. A web-based course designed for a specific group of students, like a traditional course, may not fit other students. Therefore the course material needs to be flexible so that different students may get different materials and an order of presentation that depends upon their own characteristics.

Adaptive e-learning systems (AES) try to solve these problems by altering the presentation of material to suit each individual student (Brusilovsky, 1996). Adaptive e-learning systems combine ideas from hypermedia systems and intelligent tutoring systems to adapt the systems to the particular student. They use a model of the student to collect information about his or her goals, preferences and knowledge, and use this

model throughout the interaction with the student in order to adapt to the needs of that student (Brusilovsky, 2001).

There are at least two reasons driving the advances of adaptive e-learning systems. First, e-learning applications are typically used by much more heterogeneous students than any standalone computer-based learning application. Any web-based learning system that is designed for a specific group of students may not suit other students. Second, generally the student of web based educational hypermedia is working without any assistance from teachers, as would be the case in a traditional classroom situation.

The basic components of AES are the *domain model*, the *student model* and the *adaptation model* (Cannataro, Cuzzocrea, Mastroianni, Ortale, & Pugliese, 2002). The domain model is the area or topic for which adaptive e-learning is intended as a resource. The student model is a collection of characteristics for which specific values are recorded for each student. The adaptation model is essentially a definition of what parts of the e-learning can be adapted and under what circumstances this adaptation is to occur (Pascoe & Sallis, 1998).

The main factor to provide adaptivity in the AES is the student model that represents relevant aspects of the students such as preferences, knowledge and interests (Brusilovsky, Ritter, & Schwarz, 1997). The student model dynamically maintains information for each student such as his/her knowledge, preferences, etc. The system collects this student information by observing the use of the application, by presenting series of questionnaires or feedback forms. The more accurate the student model is, the more advanced the adaptation that can be provided.

According to Huitt (2003), the learning process is complex and may be influenced by many factors including student characteristics. There are many student characteristics that are related to the learning process; among others are: prior knowledge, intelligence, study habits, age, gender, motivation, learning style, cognitive development, socio-emotional development, moral and character development. The student model should therefore accommodate these factors as much as possible in order for the AES to adapt accurately to the student's needs.

One of the current problems with AES is that most systems are capable of considering only a limited number of student characteristics for the adaptation (Carro, 2002). According to Carver, et al. (1999), current student models measure limited student characteristics which are normally restricted to a single dimension. Future

student models should take into account multiple dimensions of the student characteristics including knowledge, learning style, student goals, and preferences, etc.

One other problem faced by current AES is generality, i.e. the capability of the system to support any teaching domain. Most current AES have fixed knowledge domains which are not easily expandable or adaptable to other subject matter (Carver et al., 1999; Wu, De Kort, & De Bra, 2001). It is difficult to update teaching materials in AES or to author a new one with new subject matter (Carro, 2002). An AES should be reusable in different domains of knowledge and can be built and maintained easily (Melis et al., 2001). As AES grow in size and dimensions, the ease of construction and maintenance of the systems will increase in importance.

The two problems of the existing AES discussed above, namely: limited student characteristics and lack of genericity, show the need to develop an adaptive e-learning system that incorporates multiple student characteristics and is also subject generic.

II. Designing adaptive e-learning system

The development of an adaptive e-learning system is different in several ways from the development of general software. According to Koch (2000), these differences are mainly related to the navigation facilities, the user role, the dynamic adaptation of contents, navigation and presentation. Murugesan and Ginige (2005) also noted that the development of web application software has certain characteristics that make it different from traditional software, information systems, or computer application development. These characteristics include real-time interaction, complexity, changeability and the desire to provide personalised information.

The development process consisted of requirements analysis, design and implementation. The design models are expressed by using the Unified Modelling Language (UML). The requirements analysis is the first step in the development of the adaptive e-learning generic template. In this step, the system's overall functions need to be defined, including the major objectives and requirements of the system (Ginige & Murugesan, 2001). It is important to establish correct objectives and requirements in the early stages of the system development to prevent later errors and to ensure that the system will provide the expected functions.

The architecture of the AES is shown in Figure 1.

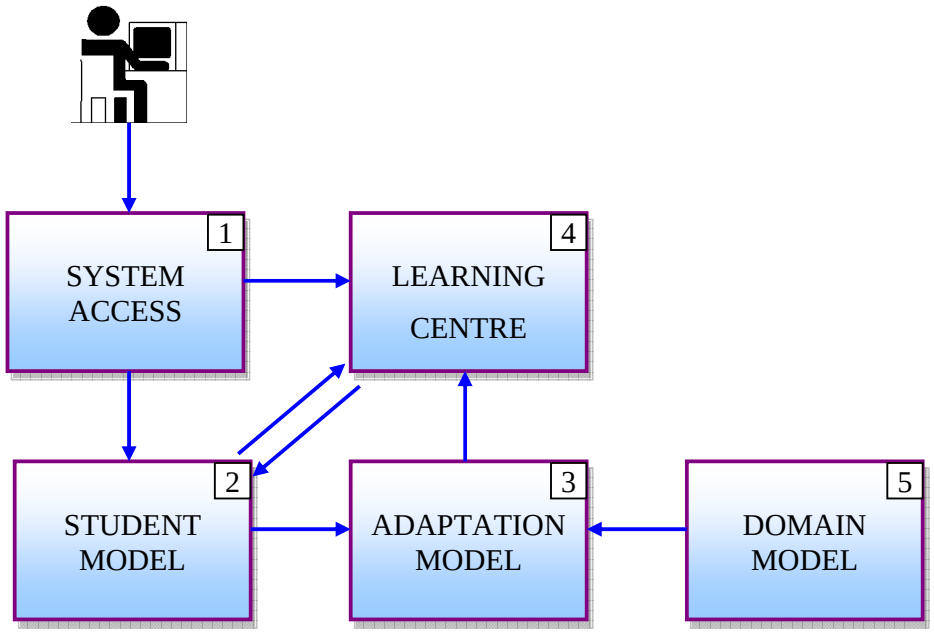

Figure 1. The AES architecture

The AES consists of five main subsystems: *System Access*, *Learning Centre*, *Student Model*, *Adaptation Model* and *Domain Model*. These subsystems are described in turn.

a. System Access

This subsystem handles registration, login and logoff functions. A student can access the AES as a registered user or an anonymous user. A registered student must login with a username and password; s/he can then logoff after a session and login again at a later time using the profile that is stored in the student model. The anonymous user does not need a username or password to login; however, s/he can continue using his or her profile as long as s/he accesses it using the same machine as before.

b. Student Model

The functions of the student model are to create, store and update the student's profile. When a new user registers or starts a new session as an anonymous user, a profile is created in the student model. The student model observes the student's activities, such as navigating through links and responding to questionnaire and tests, and then updates the profile accordingly. The current profile is used as a basis of adaptation to the learning material being presented.

c. Adaptation Model

This subsystem executes adaptation and teaching rules and provides an action to the learning centre subsystem. The adaptation rules specify how concepts or pages and links are adapted to the student. The teaching rules determine how conditions of concepts or pages and links are presented to the students. This subsystem also receives information from the domain model subsystem.

d. Learning Centre

This subsystem accommodates communication between a student and the system during his or her learning activities. The student's learning activities include learning the material, doing pre-tests and tests, filling out a questionnaire, viewing profiles, and changing the learning style (*ls*), multimedia mode (*mm*), colour, background and password. To accommodate the student's activities, this subsystem needs to communicate with other subsystems of the student model for updating and retrieving the student profile. This subsystem provides presentation to the student based on the information obtained from the adaptation model.

e. Domain Model

The domain model consists of the learning material and additional multimedia files to support the learning activities. In the AES generic template, this subsystem contains all necessary XHTML/XML files without any learning content. A teacher who wants to use the generic template needs to compose these files with his/her own learning material.

III. Implementing adaptive e-learning system

The present AES was developed as a generic template which is a fully working AES that has all adaptivity and adaptability functions but comes with empty learning material. A person with no experience in programming can use this generic template to author an AES to host his or her own learning material. All pages of the learning material need to be composed in XHTML/XML files.

The software is implemented using Java Servlets. The AES software architecture contains several modules required to accomplish necessary tasks related to the adaptivity and adaptability functions. Figure 2 shows the AES software architecture illustrating all modules and functions.

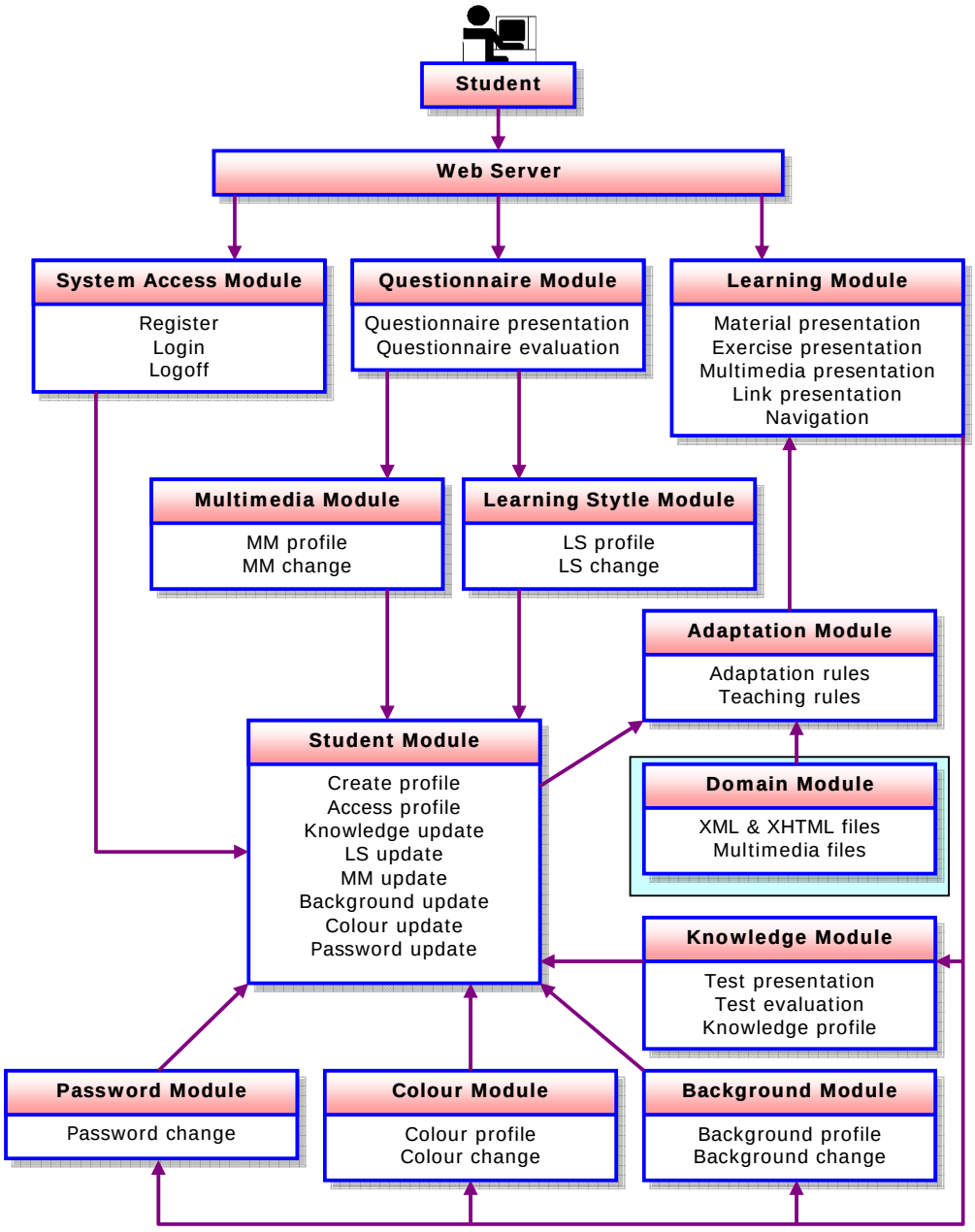

Figure 2. The AES software architecture

a. System Access Module

An access page is the first page to be accessed by students in order to enter the AES. The access page was composed in an HTML file containing some forms for registration of new users, login of registered users and entrance of new and resuming anonymous users. In addition to new users' personal data, the access page also contains some hidden data including initial values of knowledge level, learning style (*ls*), multimedia mode (*mm*) and background, all of which need to be sent to a web

server. The system access module consists of the necessary Java Servlet applications to process the submitted data and to handle the exiting session.

Having successfully entered into the system, the student is presented with a welcome page containing introductory information, links to a questionnaire, pre-test, tutorial, and access to student's profile. The welcome page, composed in XHTML and XML files, contains three columns of conditional fragments including links to questionnaire, pre-test, and tutorial and one column of fragment containing links to access the student's profile. The first three fragments were implemented using the method of *conditional inclusion of links*. The content of the conditional fragments is adapted to the level of knowledge, values of *ls* and *mm*.

b. Questionnaire, LS and MM Modules

The questionnaire page was composed using XHTML, whilst JavaScript was implemented to hide the answer choices before the question was clicked on. The student must answer all questions for *ls* and *mm* dimensions. For this purpose, JavaScript manipulation is implemented in the page so that the answers cannot be submitted before all questions have been answered.

The questionnaire answers submitted by the student are processed in the web server using a Java Servlet application of *Questioner*. There are six possibilities of the *ls* preference interpretation ranging from "very strong preference for global mode" to "very strong preference for sequential mode". A student who has a positive result (1 to 11), indicating that his or her learning style tendency is global, will be given a presentation in global mode. On the other hand, a student who has a negative result (-1 to -11), indicating that his or her learning style tendency is sequential, will be given a presentation in sequential mode.

There are six possibilities of the *mm* preference interpretation ranging from "very strong preference for non-multimedia mode" to "very strong preference for multimedia mode". A student who gets a positive result (1 to 11), indicating that s/he has verbal learning tendency, will be given a presentation of the learning material without any additional multimedia resources. On the other hand, a student who gets a negative result (-1 to -11), indicating that s/he has a tendency towards visual learning, will be presented learning material with additional multimedia resources.

The questionnaire results consisting of *ls* and *mm* dimensions are presented by the Java Servlet application in numerical and graphical values. The student's profile concerning the *ls* and *mm* values in the student model will be updated and the learning

material presentation will be adapted accordingly. The learning style and multimedia adaptation is based only on the students' responses to the questionnaire.

c. Learning Module

The purpose of the learning module is to handle student's learning-related activities in the AES, including material and multimedia presentations, link and navigation presentations. From within the learning module, the student can access other modules, such as: knowledge, background, colour and password. These modules provide the student model an update of student information related to the learning activities. The student model will then trigger the adaptation module to provide an adaptive presentation.

Presentation of the learning material is provided in either global mode or sequential mode depending on the student's *ls* values and with either additional multimedia resources or no-multimedia depending on the student's *mm* values. Based on the questionnaire result, the AES presents the learning material in one of four learning modes.

Additional multimedia resources are presented to the student who has a tendency of visual learning regardless of his or her *ls* values. The student can access navigation buttons of the multimedia features located at the bottom of the learning material page. The multimedia resources offered to the intended student include music, video clip and flash animation. Presentation of the multimedia resources through the navigation buttons is implemented using JavaScript.

The learning module also monitors student progress in a given learning activity by informing the student how many pages s/he has visited and how many more pages there are to study. This information is presented in the header of every material page. By clicking on the "visited page" link, a list of all pages that have been visited will be presented; by clicking on the "unvisited page" link, a list of all pages that have not been visited will be presented.

d. Knowledge, Background, Colour and Password Modules

The knowledge module handles some tasks related to the presentation and evaluation of tests and knowledge profile access. In order to present and evaluate the tests (which are in multiple choice formats), the module needs a Java Servlet application and an XML file. When authoring the AES for a particular domain of knowledge, the file needs to be composed to contain the questions, all choices of answers including true and false answers and an explanation for each corresponding

choice. Each time the application is executed, the questions and answers will be randomly ordered.

The same Java Servlet application evaluates the submitted answers according to the key given in the XML file and presents a feedback page to the student. The feedback page contains the student's answer, judgment whether the answer is correct or wrong, explanation why it is correct or wrong, total score obtained and recommendation links. The knowledge module also has a Java Servlet application that is used to retrieve and present the student's profile concerning his or her level of knowledge from the student model. This profile includes a percentage of the visited page for each chapter, pre-test scores, and post-test scores.

The background module enables the student to change the web page background. The module provides a list of textured pattern backgrounds that are available to choose and changes the web page background accordingly. The colour module enables the student to change colour of all types of links used within the system. The links include internal, external, active and inactive links. The module provides all possible colours to choose for links and changes them accordingly. The password module enables the student to change her or his password. Only the registered user who has a username and password can use this facility. The module does not work for anonymous users.

IV. Summary

The present AES was developed as a generic system. The development process consisting of requirements analysis, design and implementation follows an engineering approach that has advantages in terms of separating content, navigation and presentation design. The design models are expressed by using the Unified Modelling Language (UML) and the code is implemented using Java Servlets.

In the requirements analysis, all adaptivity and adaptability functionalities and features of the generic AES were defined; actors and use cases were identified; use case diagrams were built. Based on the results of the analysis, a conceptual model of the system containing all classes and objects was designed. Then, sequence diagrams were constructed. Navigation models of the system cover four main student activities including *Studying materials*, C*hanging learning mode*, *Changing knowledge level* and *Changing other profiles*. Presentation models containing a set of views that show the content and the look and feel of nodes were composed.

References

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction, 6*(2-3), 87-129.

Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User Adapted Interaction, 11*, 87-110.

Brusilovsky, P., Ritter, S., & Schwarz, E. (1997, August 18-22). *Distributed intelligent tutoring on the web.* Paper presented at the 8th World Conference on Artificial Intelligence in Education, Kobe, Japan.

Cannataro, M., Cuzzocrea, A., Mastroianni, C., Ortale, R., & Pugliese, A. (2002). *Modeling adaptive hypermedia with an object-oriented approach and XML.* Paper presented at the 2nd International Workshop on Web Dynamics (WebDyn 2002) in conjunction with the 11th International World Wide Web Conference (WWW 2002), Honolulu, Hawaii.

Carro, R. (2002, July). *Adaptive hypermedia in education: New considerations and trends.* Paper presented at the 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida.

Carver, C. A. J., Hill, J. M. D., & Pooch, U. W. (1999, October 24-30). *Third generation adaptive hypermedia systems.* Paper presented at the World Conference on the WWW and Internet (WebNet 99), Honolulu, Hawaii.

Ginige, A., & Murugesan, S. (2001). Web engineering: A methodology for developing scalable, maintainable web applications. *Cutter IT Journal, 14*(7), 24-35.

Huitt, W. (2003). A transactional model of the teaching/learning process. *Educational Psychology Interactive*, Valdosta, GA: Valdosta State University. Retrieved [19 September 2005] from http://chiron.valdosta.edu/whuitt/materials/tchlrnmd.html.

Koch, N. (2000). *Software engineering for adaptive hypermedia systems: Reference model, modeling techniques and development process.* Unpublished doctoral dissertation, Ludwig-Maximilians-Universität München, München.

Melis, E., Andr`es, E., B¨udenbender, J., Frischauf, A., Goguadze, G., Libbrecht, P., et al. (2001). ActiveMath: A generic and adaptive web-based learning environment. *Journal of Artificial Intelligence and Mathematics, 12*(4), 385-407.

Murugesan, S., & Ginige, A. (2005). Web engineering: Introduction and perspectives. In W. Suh (Ed.), *Web engineering: Principles and techniques*. Hershey, PA: Idea Group, Inc.

Pascoe, R., & Sallis, A. (1998, October). *A pedagogical basis for adaptive WWW textbooks.* Paper presented at the North American Web Developers Conference, Fredericton, NB, Canada.

Wu, H., De Kort, E., & De Bra, P. (2001, August). *Design issues for general-purpose adaptive hypermedia systems*. Paper presented at the ACM Conference on Hypertext and Hypermedia, Aarhus, Denmark.