Student Modelling in Adaptive E-Learning Systems

Vatcharaporn Esichaikul*

School of Engineering and Technology
Asian Institute of Technology, Thailand
E-mail: vatchara@ait.ac.th

Supaporn Lamnoi

National Electronics and Computer Technology Center (NECTEC)
Pathumthani 12120,Thailand
E-mail: supaporn.lamnoi@nectec.or.th

Clemens Bechter

Thammasat Business School
Thammasat University, Bangkok, Thailand
E-mail: bechter@gmail.com

*Corresponding author

Abstract: Most e-Learning systems provide web-based learning so that students can access the same online courses via the Internet without adaptation, based on each student's profile and behavior. In an e-Learning system, one size does not fit all. Therefore, it is a challenge to make e-Learning systems that are suitably "adaptive". The aim of adaptive e-Learning is to provide the students the appropriate content at the right time, means that the system is able to determine the knowledge level, keep track of usage, and arrange content automatically for each student for the best learning result. This study presents a proposed system which includes major adaptive features based on a student model. The proposed system is able to initialize the student model for determining the knowledge level of a student when the student registers for the course. After a student starts learning the lessons and doing many activities, the system can track information of the student until he/she takes a test. The student's knowledge level, based on the test scores, is updated into the system for use in the adaptation process, which combines the student model with the domain model in order to deliver suitable course contents to the students. In this study, the proposed adaptive e-Learning system is implemented on an "Introduction to Java Programming Language" course, using LearnSquare software. After the system was tested, the results showed positive feedback towards the proposed system, especially in its adaptive capability.

Keywords: e-Learning; Adaptive; Adaptation Process; Adaptation Techniques Student Model; Domain Model

Biographical notes: Vatcharaporn Esichaikul is currently an Associate Professor of at Asian Institute of Technology, Thailand. She earned her Ph.D in MIS from Kent State University. Her interests include research in digital learning and mobile learning.

Supaporn Lamnoi received her Master Degree in Information Management from Asian Institute of Technology. Currently, she is working at National Electronics and Computer Technology Center (NECTEC), Thailand.

Clemens Bechter is an Associate Professor at Thammasat Business School, Thammasat University, Thailand. He received his Ph.D. from the University of St.Gallen, Switzerland. His current research interests include cultural differences in e-Learning and simulation-based e-Learning.

1. Introduction

In the last decade, the role of information technology for education has changed rapidly and significantly with the occurrence of e-Learning systems. E-Learning systems have increased their value with the growth and popularity of computer networks through the World Wide Web (WWW) and the Internet. Combining the use of the Internet with potential teaching and learning methods offers new challenges and opportunities in distance education and e-Learning. E-Learning plays a major role in delivering educational material to the learners.

Adaptive e-Learning is a new approach that can make an e-Learning system more effective by adapting the presentation of information and overall linkage structure to individual users in accordance with their knowledge and behavior. Adaptive e-Learning is based on the assumption that each learner has different learner-characteristics and that different educational settings can be more suitable for one type of learner than for another, according to ATI, or aptitude x treatment interaction (Cronbach & Snow, 1977). When course content can be provided in a flexible way, adapted to individual learners' characteristics through the e-Learning system, the system can deliver the course content so that it capitalizes on the learner's characteristics in order to optimize the learning outcome (Brusilovsky, 1999; Brusilovsky & Peylo, 2003; Shute & Towle, 2003).

The aim of adaptive e-Learning is to provide appropriate information to the right student at the right time. This means that an adaptive e-Learning system is able to keep track of usage and to accommodate content automatically for each of the users, for the best learning result. An adaptive system is supported by a student model, which is built from a student's goals, preferences, and knowledge. Then the student model is used to adapt the interaction mode of the e-Learning system according to the user's needs (Brusilovsky, 1999; Brusilovsky & Peylo, 2003). Adaptive e-Learning systems can enhance the usability of material and thus make the e-Learning system more effective, which will improve the students' acquisition of knowledge and lead to better learning results.

2. Background on Adaptive e-Learning

The adaptation of the teaching and learning process can be divided in four elements, based on a hypothetical e-Learning system, as described below (Modritscher et al., 2004):

Adaptive content aggregation: The system can provide the students with different content types, depending on the teaching and learning style, ranging, for example, from static content to completely interactive components like simulations

or games. Moreover, the page information can be aggregated by considering the distinct background knowledge, levels of content, or types of multimedia.

Adaptive presentation: The content presentation within a page can be adapted by providing prerequisite, additional, or comparative explanations. Providing explanation variants means that the same page content can be displayed in many ways, and reordering content items would follow the relevance to background knowledge in the student model.

Adaptive navigation: The navigations or links that are presented within pages can be adapted to achieve several adaptation goals by providing adaptive navigation methods. Adaptive navigation helps the adaptation process by managing personalized views in the content pages.

Adaptive collaboration support: This technique uses a network-based educational system to form a collaborating group of learners by using the system's knowledge. It provides communication between users with the aid of a collaboration application.

There were some studies related to adaptive e-Learning but they had different focuses and approaches. Dall'Acqua (2009) proposed a multidimensional design model, describing the specifications needed for an educational environment and examined the conditions that makes a learning environment 'adaptive". Dekson and Suresh (2010) conducted a survey on the various models of adaptive content delivery system and proposed newer methods of delivering adaptive content for adaptive e-portfolio system. Recently, Mustafa and Sharif (2011) presented an approach to integrate learning styles into adaptive e-Learning hypermedia system and assessed the effect of adapting educational materials individualized to the student's learning style.

3. An Overview of Sense-Making Methodology

Adaptations in learning environment are based on well-organized models and processes. A large amount of information in adaptive e-Learning systems is needed to represent domain knowledge and to model the student learning behavior. This information can be divided into three main models: a domain model, a student model, and an adaptive model.

3.1. Domain Model

A domain model contains information about the knowledge domain of course content to support adaptive course delivery. The domain model acts as a data repository that consists of topics, contents, pages or nodes, and navigation links related to the design structure of the represented data. However, the domain model may also contain student information relevant to the learning activity, such as workflows, participants, and roles.

There are two important parts of the domain model: course content and the delivery system. The delivery system should be able to support all content types and be adaptable to different requirements of course content. The most important aspect of the domain model is the relationship between the course elements – the navigation links between pages – which are used to decide upon adaptations. The domain model focuses on designing a hypermedia structure that is appropriate for user needs and characteristics. This model designs the structures of hypermedia content to protect users from such obstacles as the "lost in space" problem when they enter each node. These content structures make it convenient for students to find the data and topics that they want.

3.2. Student Model

The main component of adaptive e-Learning systems is a student model. It is sometimes referred to as a learning model or user model. It contains all student information, for example, their domain knowledge, behavior, learning level, and other information. The student model not only collects general user information, but also maintains a live user account within the system (Paramythis & Loidl-Reisinger, 2004).

Domain-specific information and domain-independent information are two major groups of information collected in the student models, based on the relationship with the particular subject. The domain-specific information model is referred to as the *student knowledge model*. It describes the students' knowledge level, their understanding of domain knowledge or curriculum elements, the errors that the students made, the students' knowledge development process, records of learning behaviors, records of evaluation or assessment, and so forth.

The domain-independent information is information about the skills of students, so it is based on their behavior. It may include learning goals (to evaluate the learners' achievements), cognitive capabilities such as inductive reasoning skill and associative learning skill, motivational states that drive the learners, background and experience, and preferences.

3.3. Adaptive Model

An adaptive model incorporates the adaptive theory of an adaptive e-Learning system by combining the domain model with the student model. The process of adaptive modelling starts with selecting representative nodes by analyzing the student needs from the student model. Nodes can be classified into different types of knowledge: *basic knowledge*, including knowledge of definitions, formulas, and other matter; *procedural knowledge*, addressing relations among steps; and *conceptual knowledge*, referring to relations between concepts that draw details into a bigger picture (Shute & Towle, 2003). Each kind of knowledge requires different strategies, so nodes will be presented to learners in different fashions (Shute & Towle, 2003). The next step is to make a decision about which learning objects from which nodes should be represented, so that they can be used by students until they are finished with that node. The last step is to repeat the process until each node is completely selected.

4. Domain Modelling and Student Modelling

4.1. Domain Modelling

The domain model for the proposed system is divided into three parts: content hierarchy, domain content, and test construction.

Content Hierarchy

In consultation with experts, the course contents are organized into a content tree, which is similar to a table of contents in a textbook (Table 1 shows the first chapter as an example). The chapters and sections of the course material are represented by the nodes. A course is divided into chapters, sections, and subsections sequentially. There are two types of content sections: learning sections, which contain the content of learning material, and assessment sections, which contain tests or exercises. In each chapter, there

must be a test or an assessment section in the last node. A learning section may be connected to one or more concepts that are taught in the section. These content structures help students conveniently find the content and topic that they want.

Table 1. Mapping content hierarchy with course content

Chapter	Section	Subsection	Type of section	Concept
1. Introduc-tion to Java				
	1.1. What is Java?		Learning section	1.1_1 Java intro.
	1.2. Running Java Programs		Learning section	1.2_1 Bytecodes
	1.3. Small Java Programs		Learning section	1.3_1 Java structure
	1.4. How to Run the Example Programs		Learning section	1.4_1 Compile Java - Running Java
	1.5. Test1		Assessment section	

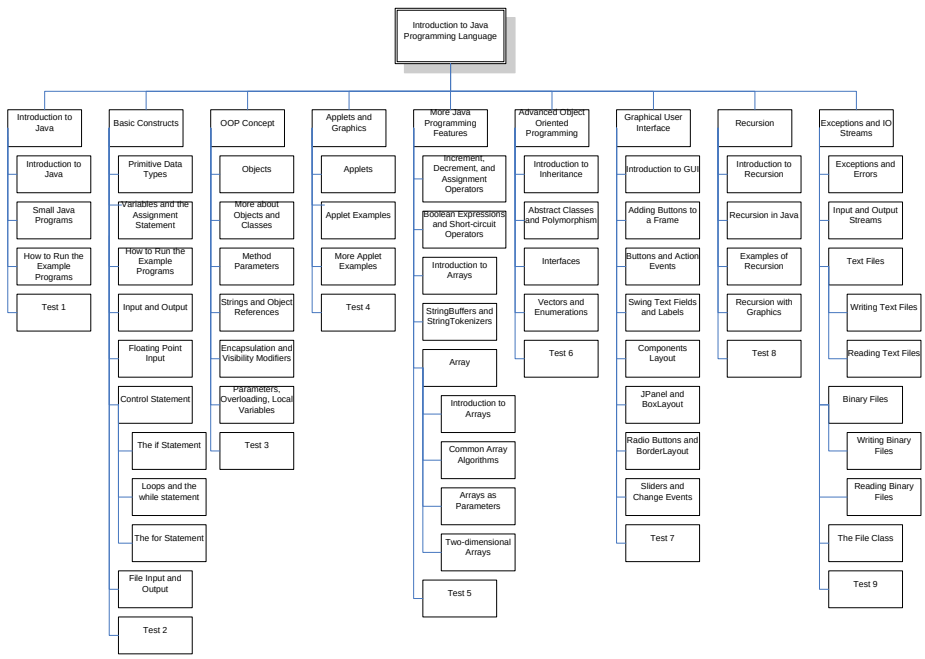

Figure 1. Content hierarchy of Java course

Domain Content

The course "Introduction to Java Programming Language" is used for the domain content case study in the proposed system. It is an online tutorial on Java programming language. The Java course is selected, prepared, and gathered into the proposed system. The teaching domain was separated into 56 sections or learning units in nine chapters. Each section was divided into 55 domain concepts and the relations between them were defined. The Java course is structured into the content hierarchy in Figure 1.

Test Construction

In the domain content structure, a test is an important element in the assessment of the student's knowledge level. It is the main factor used to adapt the course content to the student. There are several tests in one course. The proposed system includes two types of tests: a pretest, which is conducted at the beginning, after a student registers in the course, and a normal test, which is the test at the end of each chapter. The construction of both types is quite similar. Firstly, an instructor defines the general information about the test, such as the name or title of the test, the test period, etc. Then, the instructor defines the standard for assessing the learner. The standard for assessment is the "pass" percentage that the student must achieve to move to the next chapter or finish the course.

4.2. Student Modelling

There are two processes for student modelling: the initialization and update of a student model.

Initialization of Student Model

The proposed system is able to initialize the student model to determine the knowledge level of students before doing the learning activities. Figures 2.1 and 2.2 show flowcharts of the initialization of a student model and the initialization process. After the student registers for the course and logs into the proposed system, s/he is requested to take the pretest. The results from the pretest calculate the student's knowledge level.

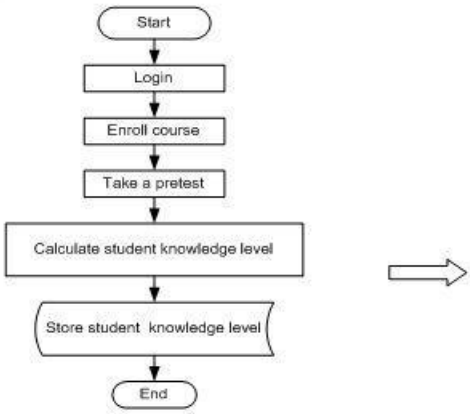

Figure 2.1 Initialization of student model flowchart

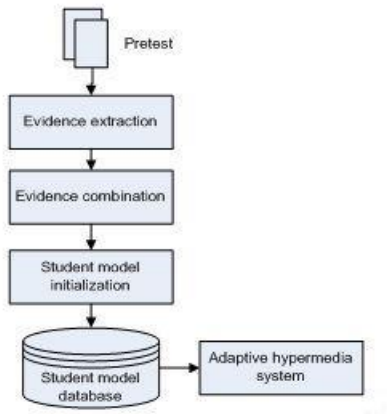

Figure 2.2 Student model initialization process

The Dempster-Shafer (DS) theory (Dempster, 1968; Shafer, 1976) is applied to analyze the results of a pretest. It is a mathematical theory of evidence based on belief function and plausibility reasoning (Sonamthiang et al., 2006). The theory was developed by Dempster and was extended by Shafer in 1976. The DS theory is used to calculate the probability of an event by combining separate pieces of evidence.

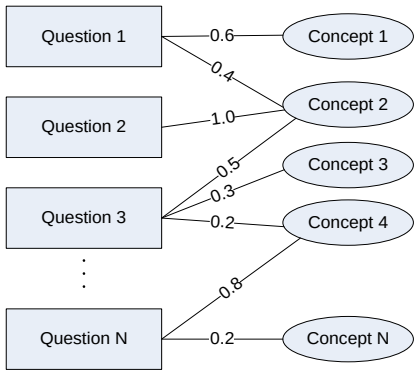

Figure 3. Relationships between questions

Figure 3 shows the relationships and weights among questions and concepts of a pretest. Usually, the relationships and weights can be determined by instructors who are familiar with the knowledge domain. Each pretest question acts as a sensor to detect the evidence. The student has to apply one or more related concepts to answer a question correctly. For example, a student must know concept1 and concept2 to answer question1 correctly. Question2 needs knowledge of concept2. Questions3 requires knowledge of concept2, concept3, and concept4. The total weight of all relationships from question to concept(s) must be equal to 1, and it is used as the mass function in the DS theory. These weights that affect the answers to questions are defined by the expert in the domain. For instance, the weight of question1 to concept1 is 0.6, and 0.4 belongs to concept2. The weights are used to calculate a student's knowledge level, depending on each concept in the domain. The pretest of this proposed system is a multiple-choice test. Each choice is used for detecting a student's understanding and is designed to classify the student's knowledge level into 3 groups: low, intermediate, and master level. Therefore, the student's knowledge levels are applied to the set of conclusions according to the following concept:

$\Theta = \{L, I, M\}$

Where L, I, and M stand for low, intermediate, and master level respectively. Therefore, the frame of discernment can be defined as:

$2^{\Theta} = \{ø, \{L\}, \{I,\}, \{M\}, \{L, I\}, \{L, M\}, \{I, M\}, \{L, I, M\}\}$

The mass function is derived from the relationships and weights of the pretest. After a student finishes the pretest, her/his choices determine her/his knowledge level. The processes to initialize the student model of the proposed system were designed as shown in Figure 2.2. The pretest is an input to create the student model. The processes are: evidence extraction, evidence combination, and student model initialization. The output of the last process is the initial student model. It is collected in the student model database to be used by other processes of the adaptive e-Learning system.

The three processes are as follows:

Evidence extraction: The student's knowledge level in every concept is acquired from this process. The pretest questions are used as a "sensor" to "perceive" the student's knowledge. The related questions determine how well the student knows the concept, with some degree of belief.

Evidence combination: Results obtained from the evidence extraction process are combined using Dempster's combination rule to find the most trustworthy conclusion. Moreover, the belief intervals are also obtained to determine the degree of uncertainty in the conclusions.

Student model initialization: The student's cognitive model uses the conclusion from each concept as an initial value. These values can be easily employed by various students' modelling. Then, the student model is stored in the student model database for further use by other processes in the adaptive hypermedia system.

An example of calculation of student knowledge level in a particular concept is explained in Table 2. If the target concept is "Control statement", then the questions that are related to this target concept are question1 and question2. The weights of all relations are defined by the instructor, as shown in Table 2.

Table 2. Relations and weights of concepts and questions

Concepts	Question 1	Question 2
1. Data types	0.3	-
2. Input & output	0.1	0.2
3. Control statement	0.6	0.5
4. Array	-	0.1
5. Exception	-	0.2

If the student answered question 1 correctly but question 2 incorrectly, this means that the student has mastered question 1, with the weight 0.6, and has low knowledge level in question 2, with the weight 0.5. Moreover, the weights of the other questions also emerge, as shown in Table 3. So Dempster's combination rule is used to determine the student's knowledge of the "Control statement" concept, as shown in the last column of Table 3. The conclusion with highest mass number, from Dempster's combination rule, is selected as the student's knowledge level.

The possible results from Dempster's combination rule are used to determine the following levels of knowledge:

L is defined as low level, or level 1.

L&I is defined as beginning level, or level 2.

I is defined as intermediate level, or level 3.

I&M is defined as advanced level, or level 4.

M is defined as mastered level, or level 5.

Table 3. Calculating student knowledge level of the "control statement" concept

Question 1		Question 2								$m_1 \oplus m_2$
		Null	L	I	M	L&I	L&M	I&M	All	
		0	0.5	0	0	0.4	0	0.1	0	
Null	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0.290
I	0	0	0	0	0	0	0	0	0	0.387
M	0.6	0	0.3	0	0	0.24	0	0.06	0	0.323
L&I	0	0	0	0	0	0	0	0	0	0
L&M	0.1	0	0.05	0	0	0.04	0	0.01	0	0
I&M	0.3	0	0.15	0	0	0.12	0	0.03	0	0.097
All	0	0	0	0	0	0	0	0	0	0

Updating Student Model

After a student logs in to the system, it can track information on the student, using the session as a key value for classifying the student. Then, the student starts to learn the lessons and carry out the activities, until s/he takes a section test. In addition, the proposed system calculates the student's knowledge level and changes its stored value if her/his level has increased or decreased. The knowledge level is calculated by using the normal test and is updated into the system for use in the adaptation process.

5. System Adaptation and Implementation

To carry out system adaptation, two elements are needed: adaptation processes and adaptation techniques.

5.1. Adaptation Processes

The adaptation processes in the proposed system combine the student model with the domain model in order to deliver suitable course content to the student. The system adapts the course contents after the student takes the pretest and the normal test.

Adaptation after Pretest

After a student enrolls in a course with the proposed system, s/he is requested to take a pretest. The results from the pretest answers are input to the DS formula for determining the level of the student's knowledge in all concepts. When the student completes the pretest, the proposed system is able to adapt the course contents based on the current knowledge level.

Adaptation after Normal Test

Further adaptation occurs after the normal test, or end of chapter test, taken once the student has finished learning all required sections. After the student finishes the normal test, the system calculates the knowledge level in all concepts and sections of the current chapter again and updates the student model database with this new value. Then, it calculates a new overall knowledge level in the same way as is done in the pretest. If the student passes an assessment of the current chapter, it offers section links to the next chapter, which should have section knowledge level greater than or equal to overall student knowledge level. If the student does not pass the assessment of the current chapter, the proposed system calculates the student's knowledge level pertaining to the concepts in this chapter again. Then, it offers section links to the current chapter that have section knowledge level greater than or equal to her/his overall knowledge level.

5.2. Adaptation Techniques

The proposed system is developed by using all techniques from adaptive navigation support technology, which are classified as direct guidance, link sorting, link hiding, and link annotation (Kavcic, 1999):

Direct guidance is the basic technique of adaptive navigation support. It is implemented by directly presenting all of the recommended links for the student to learn, with regard to her/his knowledge level in the test. It is also developed by directly telling the student in what sequence the section links should be learnt.

Link sorting is implemented by sorting all section links of a course alphabetically. The section links are sorted according to relevance to the current section node and the section knowledge levels, which are calculated from the student model. The proposed system sorts all section links in accordance with the order of links in the course. As a result, different students may see a different order of section links, depending on their section knowledge levels.

Link hiding is implemented by showing only relevant links that are suitable to the current knowledge level of the student and hiding irrelevant links from her/him. Link hiding protects the student from the complexity of all section links in a course and reduces her/his cognitive overload in hyperspace. The proposed system hides section links that have section knowledge levels lower than the student's overall knowledge level.

Link annotation is implemented by using five colored balls in front of the section links to annotate different knowledge levels of the links, so that different colors represent different knowledge levels. For example, a white ball represents a low knowledge level, an orange ball stands for a beginning level, a yellow ball stands for an intermediate level, a green ball represents an advanced level, and a blue ball stands for a mastered level.

5.3. System Implementation

This section represents the implementation of the proposed system by combining a domain model, a student model, and system adaptation.

System Architecture

The architecture of the proposed system is based on a web-based intelligent educational system. The proposed system is implemented by using a three-tier client server architecture. The client side is concerned only with user interface and connection to the server. The server side contains many scripts to process the student model and the logic of course delivery, and connect to the database. A relational database is installed in the server side for flexible distribution, updating and synchronization of course materials, and easy standardization of the interface of various tools. To implement the proposed system with a three-tier architecture, this system utilizes the following open source tools:

Clients: Firefox is used as web browser.

Server: Apache is used as web server, and PHP as server script language.

Database: MySQL is used.

Figure 4 shows the system components of the proposed system. The client-side users of this system are students and teachers/administrators. They can access the web server by using their web browser through Internet connection. A student can do many activities, such as authentication to use the proposed system, learning content materials, and assessing the proposed system. Similarly, a teacher can login to the system, observe it, and add, edit, or delete the contents. As for the server-side, it contains HTML pages of lessons and an authentication system to verify the users. It also contains an adaptive hypermedia systems (AHS) engine for adapting the content navigation to the users by using four techniques, which are direct guidance, link sorting, link hiding, and link annotation. The AHS engine uses data from all three models: the domain model, the student model, and the adaptive model. The domain model represents a lesson arranged in a content tree collected in domain DB. The student model contains personal data, test scores, and the log file of the student collected in student DB, and uses this information to initialize and update the student model. An adaptive model incorporates the adaptive theory of an adaptive e-Learning system by combining a domain model with a student model.

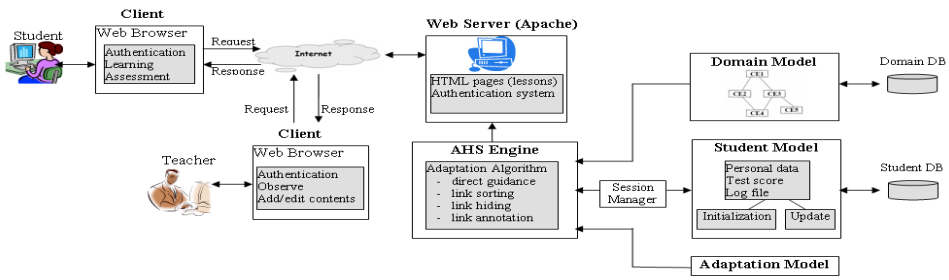

Figure 4. System components of the proposed system

LearnSquare

The proposed system is based on LearnSquare (http://www.learnsquare.com). LearnSquare is a Thai open source learning management system which is supported by the National Electronics and Computer Technology Center (NECTEC), Thailand. There are several functions in LearnSquare, such as a course construction function, an assessment function, a tracking function, and more. Moreover, LearnSquare provides a simple Content Management System (CMS), for creating contents and tests for learners. The content-creating process in the LearnSquare system starts from creating a course.

The instructor can create contents by using the editor that is provided in the proposed system or importing content that is created from outside tools. When the instructor creates a course enrollment form, students are able to enroll in the course and begin to study. The proposed system also keeps track of the studying log for each student, such as learning date and time, finished topics, duration of study, scores, and the like. The learning process ends with the instructor's determining whether the learner should pass or fail the course.

6. Evaluation

In order to gain user feedback on the proposed system, a questionnaire was used to evaluate students' opinions toward the system. The questionnaire was designed based on existing questionnaires for general evaluation of software usability (Lewis, 1995; Lewis, 2002; Davis, 1989). Additionally, the questionnaire added new items particularly related to the adaptive features of the proposed system. The questionnaire has 14 items, divided into five categories: speed of use, ease of use, user confidence, adaptive capability, and overall satisfaction with system.

The system was tested by twenty master degree students at one university. They were asked to study the course "Introduction to Java Programming Language". After the students finished the course, they were requested to complete the questionnaire. The user feedback results are shown in Table 4.

Table 4. Feedback results from proposed adaptive e-Learning system

Item	**Mean****
Speed of use	3.63
Ease of use	3.80
User confidence	3.65
Adaptive capability	4.15
Satisfaction with system	3.90

** Based on a five-point Likert scale with "strongly disagree" (1), "neutral" (3), and "strongly agree" (5)

It can be concluded that the students had a positive reaction to the proposed system, especially its adaptive capability. Further evaluation should be conducted to see the impact of this proposed system on learner performance.

7. Conclusion and Future Work

Adaptive e-Learning is an enhancement that makes e-Learning systems more effective by adapting the presentation of information and overall link structure to the individual user, based on her/his knowledge and behavior. The aim of adaptive e-Learning is to provide the right student with the right information at the right time. An adaptive system is based on three principal models: a domain model (which is all about domain content for

teaching), a student model (which collects all necessary student information), and an adaptation model (which is used in adaptation by combining both a domain model and a student model).

For student modelling, the Dempster-Shafer theory is applied to initialize and update data on the student knowledge level collected in the student model. An important aspect of this theory is its capability to manage uncertainty in scoring the student's test. It is highly accurate at determining the student knowledge level. For adaptive hypermedia, the proposed system is developed by using all techniques from adaptive navigation support, which are direct guidance, link sorting, link hiding, and link annotation. The proposed system can recommend and sort all section links that are suitable for the student's current knowledge level. Furthermore, it can annotate the section knowledge level with colored balls to represent various grades.

In addition, the system was evaluated by users to obtain feedback on the proposed system in terms of speed of use, ease of use, user confidence, adaptive capability, and overall satisfaction with the system. The results show positive opinions towards the proposed system, especially in its adaptive capability.

The major directions for future work regarding the system feature include the following:

- Implementing an adaptive e-Learning system based on the SCORM standard in order to share the course contents with another e-Learning system.
- Combining the proposed system with another adaptive hypermedia technology, for example, an adaptive presentation to represent the course contents differently and improve system usability in terms of personalization.
- Making adaptive tests or assessments more effective at adapting the difficulty level of the questions to the current knowledge level of the student.
- Adding more student information into the student model, for example, the student's learning styles, behavior, goals, preferences, backgrounds, and experience, so as to be used in the adaptation processes.

References

1. Brusilovsky, P. (1999). Adaptive hypermedia: from intelligent tutoring systems to web-based education. *Künstliche Intelligenz*, 4, 19-25. http://www2.sis.pitt.edu/~peterb/papers/KIreview.html.

2. Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education, 13*, 156-169.

3. Cronbach, L.J., & Snow, R.E. (1977). *Aptitudes and Instructional Methods: A handbook for research on interactions.* New York: Irvington.

4. Dall'Acqua, L. (2009). A model for an adaptive e-Learning environment. *Proceedings of the World Congress on Engineering and Computer Science (WCEES).*

5. Davis, F.D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319-340.

6. Dekson, D.E., & Suresh, E.S.M. (2010). Adaptive E-Learning techniques in the development of teaching Electronic Portfolio – A survey. *International Journal of Engineering Science and Technology*, *2*(9), 4175-4181.

7. Dempster, A.P. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society, Series B, 30,* 205-247.

8. Kavcic, A. (1999). Adaptation techniques in adaptive hypermedia systems. *Proceedings of the 22nd International Convention MIPRO 1999*, Conference on Multimedia and Hypermedia Systems.

9. Lewis, J.R. (1995). IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, *7* (1), 57-78.

10. Lewis, J.R. (2002). Psychometric evaluation of the PSSUQ using data from five years of usability studies. *International Journal of Human-Computer Interaction, 14*, 463-488.

11. Modritscher, F., Gutl, C., Garcia B., & Maurer, H. (2004). *Enhancement of SCORM to support adaptive e-learning within the scope of the research project AdeLE*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.6119&rep=rep1&type=pdf.

12. Mustafa, Y.E.A., & Sharif, S.M. (2011). An approach to adaptive E-Learning hypermedia system based on learning styles (AEHS-LS): implementation and evaluation. *International Journal of Library and Information Science*, *3*(1), 15-28.

13. Paramythis, A., & Loidl-Reisinger, S. (2004). *Adaptive learning environments and eLearning standards*. http://www.fim.uni-linz.ac.at/staff/paramythis/papers/ecel2003.pdf.

14. Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

15. Shute, V. & Towle, B. (2003) Adaptive e-learning. *Educational Psychologist*, *38*(2), 105-114.

16. Sonamthiang, S., Naluedomkul, K., Cercone N., & Sirinaovakul, B. (2006). Initializing student models using Dempster-Shafer theory. *Proceedings of International Conference on Computer and Advanced Technology in Education (CATE2006)*.