TOWSON UNIVERSITY

COLLEGE OF GRADUATE STUDIES AND RESEARCH

An Intelligent and Effective E-learning System That Provides Tailored Lessons to Students

By

Dehui Li

A Dissertation

Presented to the faculty of

Towson University

In partial fulfillment

Of the requirements for the degree

Doctor of Science

August 2016

Towson University

Towson, Maryland 21252

ACKNOWLEDGEMENTS

The department of computer science at Towson University has been the context of my academic life for the past 5 years. I am deeply grateful for its intellectual climate and financial support, which made this research possible.

My early enthusiasm in artificial intelligence was nurtured in classes taught by Dr. Harry Zhou who generously provided his expertise, time and advice ever since. As my advisor, he helped me to complete my education to the Ph.D. level.

I would like to express my special thanks to my program director, Dr. Chao Lu, for his help， Consideration and encouragement during the development for this thesis.

I am also indebted to the other member of my committee, Dr. Song, Yeong-tae, Dr. Feng, Jinjuan, and Dr. Wang, Yuanqiong (Kathy), for their helpful comments and discussions.

Finally, my most sincere thanks go to my family. This dissertation is dedicated with love and gratitude:

To Feng Wang, my wife, for reasons deep and private.

To Baorong Li and Qibo Ma, my parents, for their support，understanding and influence.

TOWSON UNIVERSITY
COLLEGE OF GRADUATE STUDIES AND RESEARCH

DISSERTATION APPROVAL PAGE

This is to certify that the dissertation prepared by Dehui Li, entitle "**An Intelligent and Effective E-learning System (IELS)**", has been approved by this committee as satisfactory completion of the requirement for the degree of Doctor of Science in Information Technology in the department of Computer and Information Sciences.

Dr. Harry Zhou
Chair, Thesis Committee

8-16-2016
Date

Dr. Yeong-tae Song
Committee Member

8-16-2016
Date

Dr. Kathy Wang
Committee Member

8-16-2016
Date

Dr. Jinjuan Feng
Committee Member

8-16-2016
Date

Dr. Chao Lu
Chair, Department of COSC

8/16/2016
Date

Dr. Ramesh Karne
Doctoral Program Director

8-16-2016
Date

Dr. Janet V. Delany
Dean of Graduate Studies

11-1-16
Date

3

Table of Contents

Table of Contents - Continued

Table of Contents - *Continued*

LIST OF TABLES

LIST OF FIGURES

An Intelligent and Effective E-learning System That Provides Tailored Lessons to Students

1. Introduction

1.1 Background of the Problem

'Once we free ourselves from the mental limits of viewing this technology as a weak sister to face-to face synchronous education, the potentials to revolutionize education and learning become readily apparent- Turoff', 1995.

The traditional teaching environment is usually thought to be that of a classroom: a single teacher giving lectures to a group of students who are expected to use their notes and textbook to prepare for periodic examinations and demonstrate that they have learned. An obvious problem with this approach is that everyone receives the same lecture within a fixed time frame. With so many students with different levels of understanding, it is impossible for the teacher to provide tailored lessons to every student.

Most of the E-learning systems lack artificial intelligence and merely present the content materials without evaluating the students' comprehension and competence. The lecture materials in traditional e-learning systems are presented in a predefined order and within a certain timeframe regardless the students' understanding of the topic being discussed. If some students did not understand the materials, all they can do is to repeat the same materials all over again. These E-learning systems cannot handle a large and potentially diverse student population.

1.2 Purpose of the Study

In responses to these challenges and difficulties, this dissertation proposes designs, implements, and tests An Intelligent and Effective E-learning System (IELS) that provides individualized lessons to students based on their levels of comprehension, progress and weakness. IELS combines expert knowledge, analogical reasoning and fuzzy reasoning to provide tailored lessons to students. By analyzing the student's statistic data on his/her background and intellectual ability, and dynamic data collected during a lecture session in real time, IELS is able to provide personalized lessons with different levels of difficulty for students with diverse backgrounds. IELS evaluates the student's real time learning activity, determines their competency level, analyzes their progress, and selects appropriate teaching materials. Good students can finish a lecture unit much faster than others, while the students at the introductory level may take longer. All students, hopefully, can meet the lecture objective at the end of a lecture. A variety of students with different backgrounds and abilities can benefit from this effective, efficient and individualized pedagogical strategy. The knowledge base in IELS captures the expertise of domain subject experts and uses it to dynamically construct a lecture content based on the student's competency. The case base in IELS enables it to recognize similar situations and recall and adapt its past course content for students with similar characteristics. The fuzzy reasoning component allows IELS to conduct approximate reason and handle vague and imprecise terms. By combing expert's knowledge, analogical reasoning and fuzzy reasoning, IELS demonstrates its adaptive ability to deliver personalized courses to students. To show its benefits and feasibility, IELS has been tested in the domain of computer science courses, but its design and structure promise to be domain-independent. Without any structure changes, any domain subject expert, such as in the fields of SAT, GRE, MCAT or any college courses, can input their lectures with ease. The potential applications of IELS are promising and unlimited.

In this dissertation we will give a brief description on the System design, major component, and experiment of IELS.

2. Literature Review

2.1 Adaptive E-Learning

The intent of an adaptive learning system is to provide a personalized learning resource for students, especially tailoring of learning content and offering user-preferred interfaces for processing their knowledge acquisition (Aroyo et al., 2006). Brusilovsky (2001) has indicated that in developing web-based adaptive learning systems there are two adaptation approaches employed. The first "adaptive presentation" delivers personalized content for individual students. The second "adaptive navigation support" guides individuals to access the educational content through suggestion of personalized learning paths. Other researchers have further indicated the importance of providing personalized user interfaces to adapt to and more closely meet the learning habits of students (Mampadi, Chen, Ghinea, & Chen, 2011).

In the past decade, various adaptive learning systems that represent the characteristics or preferences of students as well as the attributes of learning content (Wang & Wu, 2011) have been developed based on different parameters. For example, Karampiperis and Sampson (2005) proposed an adaptive resource selection scheme by generating all of the candidate learning paths that matched the learning objectives and then selecting the most fitting one based on the suitability of the learning resources for the individual students. Hwang, Kuo, Yin, and Chuang (2010) further developed an adaptive learning system to guide individuals to learn in a real-world environment by generating the personalized learning paths based on the learning status of each student and the relationships between the authentic learning goals and objectives. These diverse adaptive learning system efforts indicate that the provision of personalization or adaptation modules, including personalized learning materials, navigation paths or user interfaces, is recognized as an important component for developing effective learning systems (Chiou, Tseng, Hwang, & Heller, 2010; van Seters, Ossevoort, Tramper, & Goedhart, 2012).

There have been several studies conducted to develop adaptive learning systems based on learning styles or cognitive styles. For example, Tseng, Chu, Hwang, and Tsai (2008) proposed an adaptive learning system for elementary school mathematics courses by considering the different students' learning styles and the difficulty of mastering the learning content. Mampadi, Chen, Ghinea, and Chen (2011) developed a web-based learning environment that provided different user interfaces based on students' cognitive styles. Furthermore, Hsieh, Jang, Hwang, and Chen (2011) developed an adaptive mobile learning system that guided individual students to learn in a butterfly ecology garden based on students' learning styles. However, few studies have integrated multiple learning criteria, including learning styles, cognitive styles, and assessment of knowledge levels, in the development of an adaptive learning system.

In addition, Kulik and Kulik (1991), Bangert-Drowns et al. (1985), and Baxter (1990) have defined several types of computer-aided education systems:

- Computer-Assisted Instruction systems provide drill and practice exercises and tutorial instruction.
- Computer-Managed Instruction systems evaluate and store student performance and guide students to appropriate instructional resources.
- Computer-Enriched Instruction systems satisfy student requests such as solving a mathematical equation, generating data, and executing programs.

As summarized in Graesser et al. (2001), intelligent tutoring systems (ITS) are clearly one of the successful enterprises in artificial intelligence (AI). There is a long list of Intelligent Tutoring Systems (ITS) that have been tested on student populations and have proven to facilitate learning. These ITS use a variety of computational modules that are familiar to those of us in AI: production systems, Bayesian networks, schema templates, theorem proving, and explanatory reasoning. Graesser et al. (2001) also pointed out the weaknesses of the current state of tutoring systems: First, it is possible for students to guess and find a correct answer and this shallow learning is not detectable by the system. Second, ITS do not involve students in conversational exchanges so students may fail to learn the domain's language. Third, to facilitate understanding of the students' thinking, the graphical user interface (GUI) of the Intelligent Tutoring Systems (ITS) promote "focus and interaction" with the learning content details instead of improving student grasp of the overall picture of a solution.

2.2 Existing Intelligent Tutoring Systems (ITS)

The first generation of computer assisted education tools were called, Computer-Aided Instruction (CAI) systems. One of the early examples of such a tutoring system is the 'Uhr' system in the year 1969 (Sleeman & Brown, 1982). This system generated problems on arithmetic and questions on vocabulary. However, the main shortcoming of this system was it lacked user modeling or adaptive technique.

Some contemporary computer-aided systems, (Suppes et al., 1967, Woods & Hartley, 1971, Sleeman & Brown, 1982) could be called adaptive because the problems are presented in response to the user's performances. The user model employed is quite primitive in nature. The model is a parametric summary without storage of the actual knowledge state of the user. These computer-aided systems can be termed as the precursor to Intelligent Tutoring System (ITS).

In the meantime, another genre of tutoring system evolved. These types of systems were called Drill and Test as only problems were presented to the students in form of tests. The system provides the students with the test results. A simple variation of this system was the Adaptive Drill and Test. In this variety instead of just presenting the problems, the student's performance and response were collected, tabulated, and later used to select future problems. This adaptive innovation acknowledged that the students' needs was an important factor and predetermined rules would not work for all learners. This advancement recognized an adaptation technique was required to address all possible responses from the students.

The premise of the two-sigma problem states that students who receive one-on-one instruction perform two standard deviations better than students who receive traditional classroom instruction (Bloom, 1984). It is impossible for any institution to provide personal teachers to interact with each student. This limitation strongly supports the use of computers, as a replacement for one-on-one instruction from human teachers. Motivated by these reasons, there has been a surge of research groups working in this field developing various systems with various features. In 1982, Sleeman and Brown reviewed the state of the art in computer-aided instruction and first coined the term Intelligent Tutoring Systems (ITS) to describe these evolving systems and distinguish them from the previous CAI systems. They (Sleeman & Brown, 1982) defined an ITS as being:

- Computer-based
- Problem-solving
- Monitors
- Coaches
- Laboratory instructors
- Consultants

For the first time the use of Artificial Intelligence (AI) was established, which made the systems adaptive, responsive, and intelligent. With new AI techniques emerging it seemed that the computers were almost capable of processing information and thinking like humans. This motivated further ITS research. The application of AI in ITS made it possible to more easily achieve the instructive goals.

One Intelligent Tutoring Systems (ITS) is based on the teaching methodology proposed by Fissilis (et al., 1996). After studying a certain unit, the intelligent tutor dynamically creates a test. If the student passes the test, they go on to the next unit; otherwise, the system provides them a remedial unit with its corresponding test. If the student fails the test, the remedial presentation process continues until the learner fails all of the remedial units, at which point the student would repeat the process from the initial remedial unit. If the student passes the remedial unit, they will return to the lesson where they committed the original error to repeat the test they failed. At the end of the course, the students are tested on all of the learning units so that the teacher can grade their understanding. This educational process provides the student with a certain degree of flexibility to either navigate back to study the units again and/or take the corresponding tests.

Andes (Conati et al, 2002; Gertner & VanLehn, 2000) is an Intelligent Tutoring Systems (ITS) which was developed to teach physics to Naval Academy students. Andes primarily used Bayesian networks for decision-making on presentation of learning content. The major foci of the system are:

- Select the most suitable strategy for each student
- Predict the students' actions
- Perform a long-term assessment of the students' domain knowledge

Andes is a domain dependent ITS. Each problem in the system was broken into several steps and using those steps as nodes the Bayesian network was formed. Each problem in the system was represented in Bayesian networks and would predict the most probable path for the student during a course. While each student could have different approaches to a problem, the network would adjust by calculating and analyzing the changed probabilities. For a new problem, Andes predicted the best strategy for the student. There is also a problem-solver in the system that partially or wholly solved a problem to assist the students' understanding. Each Bayesian network consisted of a static and dynamic component.

The static part features Rule nodes and Context-rule nodes. The Rule node represented general physics rules with binary values, T for true and F for false. The probability P (Rule=T) was the probability the student could apply the rule properly in any situation. Initially these prior probabilities had 0.5 values but the authors claimed more realistic values would be obtained after a trial run in the naval academy.

The dynamic part contained the Context-rule node along with Fact, Goal, Rule-application, and Strategy-nodes. The Fact and the Goal nodes were also binary. P (Fact=T) was the probability that the student knew the fact and P (Goal=T) was the probability that the student is pursuing the goal. There might be more than one way to reach the Goal or the Fact nodes and that led to a large volume of parents. The conditional probabilities P (Fact=T| parenti) represented the probability of reaching the fact from parenti. The Strategy nodes existed where the students could choose from more than one option. The Rule-application nodes represent the children of the Strategy nodes. The Rule-application nodes represented the different applications of the Strategy nodes. The student could choose only one strategy at a time. Thus if the evidence increased the probability of one of the strategies it would inversely decrease the probability of the other strategies. The probability values (P (Strategy-node = {x: x ε {child1… childn}}) of these Strategy nodes would depend upon the number of children, or Rule-application nodes associated to the strategy node.

Finally, the Rule-application nodes were the connectors between Context-rule nodes, Strategy nodes, Fact and Goal nodes to new derived Fact and Goal nodes. In other words, those nodes had a single Context-rule node and Strategy node and one or more than one Fact and Goal nodes as preconditions or parents, and the children or conclusions of those nodes included some Fact and Goal nodes. They had binary values and P (R-A=T) meant the probability of a student applying the parent Context-rule to the preconditioned Fact and Goal nodes to get the derived Fact and Goal nodes. The probability values would vary with students and thus application of rules, the selection from alternate paths would depend upon each student. How Andes derived the probabilities was not stated.

SQL-Tutor (Wang & Mitrovic, 2002; Mitrovic, 2003) is an Intelligent Tutoring Systems (ITS), which as the name suggests was developed to teach university students SQL. An Artificial Neural Network (ANN) was used in SQL-Tutor for decision-making. The ITS used an ANN to model an agent that would analyze the student attributes to select an appropriate problem from the database. In SQL-Tutor the solutions to the problems were represented in the system as constraints. Whenever a student submitted a solution, the system calculated the correctness by comparing the number of constraints violated by the student. The selection of the next problem or any other required teaching decision depended on the number of mistakes or constraints the student had violated. The ANN was a feed forward network with four inputs, one output, and a hidden layer.

Delta-bar-delta back propagation and linear tanh (LT) transfer function was used and the inputs consisted of different student data points:

- Time needed to solve the problem
- The level of help provided to the student
- The complexity of the problem
- The knowledge level of the student

In the output, the ANN attempted to predict the number of errors or constraints violations committed by the student. The system employs this prediction to make the next teaching decision such as selecting the next problem from the problem database. How the weights of the ANN were chosen and exactly how the ANN was trained was not clearly explained. The authors claimed that the ANN could predict the number of errors with an accuracy of 98.06%. An added advantage of this system was it provided feedback to the student after checking the solution. The feedback might contain hints, offer a partial solution, or detail a complete solution as required.

C++ Tutor. C++ Tutor is a rule-based Intelligent Tutoring Systems (ITS) (Baffes & Mooney, 1996). The system explained the concepts of C++ using some rules. These rules were in form of Horn sentences and were called the Theory. The problems were generated to the students in the form of feature vectors. Choosing from a set of labels the students were supposed to label the vector. An algorithm called NEITHER took these labeled vectors of the student‘s solution as input and modified the correct rule base, so that the modified rules implied the student‘s solution rather than the correct solution. This process is called Theory-revision. So now the modified rule base reflected the student ‘s state of understanding, representing the student ‘s correct knowledge as well as their misconceptions. After the Theory-revision was complete the system tried to explain the errors in the student ‘s concept by showing examples, which enumerated the areas where the student’s solution had gone wrong. C++ Tutor did this automatically by comparing the modified rules with the correct ones.

PACT (Koedingeret al. 1997) is an intelligent tutoring system for algebra problem solving. PAT stands for PUMP Algebra Tutor or Practical Algebra Tutor. PAT was built to support this kind of mathematical investigation and problem solving. Most importantly, PAT was designed to help students develop algebraic skills, which they can use in the context of real-life problem situations.

The PAT learning environment includes a set of computational tools to aid investigation. The spreadsheet, graphic, and symbolic calculator augmented the organized curriculum of problem situations. The design of PAT was also guided by theoretical principles. As a cognitive tutor (Anderson, et. al, 1995), PAT has the defining feature of containing a psychological model of the cognitive processes behind successful and near-successful student performance. Based on the ACT theory, this cognitive model is written as a system of if-then production rules that are capable to generate the multitude of solution steps and errant steps typical of students.

The cognitive model is the basis for two student modeling techniques: model tracing and knowledge tracing. Model tracing is used to monitor student's progress through a problem solution (see Anderson, Boyle, Corbett, & Lewis, 1990). This tracing is done in the background by matching student actions to those the model might generate. The tutor is mostly silent until the student requires help. Then the tutor knows where the student is within the course and can provide hints that are individualized to the student's approach to the problem. PACT employs knowledge tracing to monitor students' learning from problem to problem (Corbett & Anderson, 1992). A Bayesian estimation procedure identifies students' strengths and weaknesses relative to the production rules in the cognitive model.

The Tutoring Research Group (TRG) at the University of Memphis developed Auto Tutor to simulate the dialogue patterns of typical human tutors (Graesser et al. 1999; Person et al. 2001). Auto Tutor tries to comprehend student contributions and simulate dialogue moves of either normal, unskilled tutors or sophisticated tutors. Auto Tutor was developed for college students who are taking an introductory course in computer literacy.

Instead of merely being an information delivery system, Auto Tutor is a collaborative scaffold that assists the student in actively constructing knowledge by holding a conversation in natural language. A dialog manager coordinates the conversation that occurs between a learner and a pedagogical agent, whereas lesson content and world knowledge are represented in a curriculum script and latent semantic analysis (Landauer, Foltz, & Laham, 1998).

SAM (Cassell et al., 2000; Ryokai, Vaucelle, & Cassell, 2002) is an example of a virtual peer construct. Virtual peers are a unique kind of pedagogical agent, which Cassell designed in 2000. Whereas the vast majority of pedagogical agent research is modeled after teachers or tutors (Lester, Voerman, Towns, & Callaway, 1997), virtual peers exist as a playmate and learning companion, in line with the literature on children's development in a peer context.

Virtual peers are a kind of embodied conversational agent (Cassell, Sullivan, Prevost, & Churchill, 2000), which means that they can have conversations with real people, using language but also hand gestures, facial expressions, eye gaze, and other types of "body language." Virtual peers are designed using a unique methodology that relies on data about children's natural development to build technologies that can emulate the role of a friend or playmate.

The SAM virtual peer looks like a child who is sitting behind a toy castle and waiting for a playmate to tell stories with. To this end, the SAM system has two components: an embodied conversational agent that is a life-sized child named Sam and a toy castle with several plastic figurines. Sam is projected on a screen behind the castle, and can both tell stories, using a recorded child's voice, and listen to the real child's stories, responding with appropriate feedback and short comments.

2.3 Existing ITSs with CBR Concept

There are Intelligent Tutoring Systems (ITS) that use CBR or machine learning. For example, Weber and Brusilovsky (2001) described the ELM Adaptive Remote Tutor (ELM-ART) that supports learning programming in LISP.

ELM-ART models individual learners as a collection of episodes that describe how exercise problems have been solved by a particular student. Each episode contains all the concepts and rules needed to produce the program code the students offered as solutions to programming tasks.

Each episode is stored as a case, with each case describing a concept and a rule used to solve a plan or sub-plan of the programming task. Using a combination of an overlay model and the above episodic student model, ELM-ART provides adaptive navigation support, course sequencing, individualized diagnosis of student solutions, and example based problem-solving support.

ELM-ART also selects the best next step for a particular user. Starting from the current learning goal, the system recursively computes all prerequisites that are necessary to fulfill the goal. The first concept belonging to the set of prerequisites that is not already learned or solved is selected and presented to the learner. The learner completes the course successfully when all current goal prerequisites are fulfilled and no additional goal can be selected. The system also provides a sequence of help messages with increasingly detailed explanation of the error or suboptimal solution as learner feedback.

The sequence begins with a very vague hint on what is wrong and ends with a code-level suggestion of how to correct the error or complete the solution. The system also provides an ordered list of relevant examples. Using case based retrieval; ELM-ART selects the episodes with the highest similarity values to the current frame and presents the list of links to examples and reminders. However, ELM-ART used neither adaptation nor learning. Thus, significant development and instructor effort must be invested to ensure the quality of the cases.

ActiveMath. Melis et al. (2001) provided a comprehensive account of ActiveMath, a generic web-based learning system that dynamically generates interactive mathematical courses adapted to student goals, preferences, capabilities, and knowledge. When the user has chosen goal concepts and selected a scenario, the session manager sends this request to the course generator. The course generator is responsible for choosing and arranging the learning content. It checks the user model to find out the user's prior knowledge and preferences and uses pedagogical rules to select, annotate, and arrange the content, examples, and exercises. The Active Math course generator produces the entire tutorial together with the examples and exercises and does not consider the real-time interactivity between the user and the examples or exercises.

In the CBIP project, Elorriaga and Fernández-Castro (2000) integrated a case-based instructional planner with existing Intelligent Tutoring Systems (ITS) to enhance the pedagogical component with learning capabilities. This transformed the ITS into self-improving systems that learn from memorization and from their own experiences, where instructional planning is the process of mapping out a global sequence of instructional goals and actions that provides consistency, coherence, and continuity throughout an instructional session. In CBIP, the instructional plan memory (IPM) is the repository of the past teaching and learning experiences of the case-based system.

A case defines a piece of a previously used instructional plan and includes the:

- Context in which the instructional plan was applied
- Instructional plan itself or a component part of it where the plan is layered
- Results that the instructional plan achieved

Similar to the design described in this paper, the CBIP application context consists of a sequence of student-related, session-related, and domain-related features.

In an overview on the state of adaptive e-learning systems, (Azough, Bellafkih, Bouyakhf, 2010), the multitude of advantages are explored. Particularly, web-based ITS enables the user access to an engaging variety of learning resources including images, audio, video, and graphical simulations. The ease of professional production assures the resources are of good quality and can be assembled from diverse contributing experts. E-learning systems exploit these resources in order to organize learning experiences that are flexible and available whenever the users' have time in their schedules. Nevertheless, the majority of the existing formation platforms are generally conceived as content distribution systems, without facility to address the interests of singular learners. Actually, the concept of e-learning systems suffers from several inherent lacks: the teacher is not permanently present, the teacher is not directly interacting with the learner; the management of the immediate reactions of the learner is difficult.

From the idea of 'evaluating an intelligent tutoring system for design patterns' the DEPTHS provided students with the benefits of one-on-one instruction in a cost effective manner. While it was primarily intended for teaching undergraduate students of

Computer Science, it can be equally successful in other education settings, as well. DEPTHS perform adaptation of the teaching material based on the student performance and the usage tracking data collected during the learning session. The quality of the adaptation provided by the system depends on many factors, such as the accuracy of the diagnostic tools that collect and process data, the student model that is used to store the data and the embedded teaching rules. The only way to verify the quality of the DEPTHS's adaptation functionalities is to evaluate the system in real conditions, with students who are learning the subject material.

In a "personalized e-learning system using three parameters and genetic algorithms", the curriculum sequencing is one of most appealing challenges in web-based learning environment. The success depends on the system capability to adapt the learning material automatically to the student's educational needs to promote learning performance. According to expert views there are no fixed learning paths appropriate for all learners. Various approaches to sequencing have been explored in numerous ITS projects.

2.4 Learning and Cognitive Styles in Intelligent Tutoring Systems (ITS)

Learning styles have been recognized as being an important factor for better understanding the model of learning and the learning dispositions/preferences of students (Filippidis & Tsoukalas, 2009). Keefe (1987) defined an individual's learning style as a consistent way of functioning that reflects the underlying causes of learning behaviors. Keefe (1991) pointed out that learning style is a student characteristic indicating how a student learns and likes to learn. He also stated that learning style could be an instructional strategy informing the cognition, context and content of learning. Reiff

(1992) indicated that learning styles are likely to influence how students learn, how instructors teach, and how they interact. Coffield, Moseley, Hall, and Ecclestone (2004) further suggested that teachers and course designers pay attention to students' learning styles and design teaching and learning interventions accordingly.

There have been several learning style theories proposed by researchers, such as those proposed by Honey and Mumford (1992), Keefe (1979), Kolb (1984) and Felder and Silverman (1988). Several previous studies have demonstrated the use of learning styles as one of the parameters for providing personalized learning guidance or contents (Graf, Lin, & Kinshuk, 2007; Papanikolaou, Mabbott, Bull, & Grigoriadou, 2006; Tseng, Chu, Hwang, & Tsai, 2008). Among various learning styles, the Felder–Silverman Learning Style Model (FSLSM) developed by Felder and Soloman (1997) have been recognized by many researchers as being a highly suitable model for developing adaptive learning systems (Huang, Lin, & Huang, 2012; Akbulut & Cardk, 2012).

Carver, Howard and Lane (1999) indicated that FSLSM could be the most appropriate measurement for developing hypermedia courseware by taking into personal factors into account. Kuljis and Lui (2005) further compared several learning style models, and suggested that FSLSM is the most appropriate model with respect to the application in e-learning systems. Consequently, this study adopted FSLSM as one of the factors for developing the adaptive learning system.

On the other hand, cognitive style has been recognized as being a significant factor influencing students' information seeking and processing (Frias-Martinez, Chen, & Liu, 2008). It has also been identified as an important factor impacting the effectiveness of user interfaces and the navigation strategies of learning systems (Mampadi, Chen, Ghinea, & Chen, 2011). Several studies have shown the effectiveness of considering cognitive styles in designing user interfaces for information seeking (Frias-Martinez, Chen, & Liu, 2008) and developing adaptive learning systems for providing personalized learning guidance (Evans, & Waring, 2011; Lo, Chan, & Yeh, 2012).

Among various proposed cognitive styles, the field dependent (FD) and field independent (FI) styles proposed by Witkin, Moore, Goodenough, and Cox (1977) are the most frequently adopted. Several studies have reported the usefulness of FI/FD cognitive styles in determining the suitability of learning supports or learning system designs (Gerjets, Scheiter, Opfermann, Hesse, & Eysink, 2009; Lin, Hwang, & Kuo, 2009). For example, Weller, Repman, and Rooze (1995) indicated that FI/FD cognitive style is very suitable for personalized learning design since it reveals how well a learner is able to restructure information based on the use of salient cues and field arrangement. Ford and Chen (2000) further indicated that the FD/FI cognitive style is highly related to hypermedia navigation and is very suitable for evaluating the usability of websites to students. Therefore, in this study, FI/FD cognitive style is adopted as another factor for developing the adaptive learning system.

Scholars have proposed different aspects to address the relationships between learning styles and cognitive styles. For example, some scholars have indicated that learning styles are applied cognitive styles (Keefe, 1979; Jonassen & Grabowski, 1993; Papanikolaou, Mabbott, Bull, & Grigoriadou, 2006); some have further concluded that learning styles could be viewed as a subset of cognitive styles, and could be classified as activity-centered cognitive styles (Huang, Lin, & Huang, 2011). However, the common definition of cognitive style refers to the individual differences in preferred ways of organizing and processing information and experience (Chen & Macredie, 2002; Triantafillou, Pomportsis, & Demetriadis, 2003), while learning style is defined as a consistent way of functioning that reflects the underlying causes of learning behaviors (Keefe, 1987). Moreover, cognitive styles deal with a cognitive activity such as thinking, perceiving, and remembering, while learning styles are indicators of how learners perceive, interact with and respond to learning environments, including cognitive, affective and psychological behaviors (Triantafillou, Pomportsis, & Demetriadis, 2003).

To deal with the relationship between cognitive and learning styles, researchers have indicated that cognitive styles could be classified as cognition centered, personality centered, or activity centered while learning style can be perceived as the activity-centered cognitive style (Sternberg & Grigorenko, 1997). From this aspect, learning styles are viewed as a subset of cognitive styles (Riding & Rayner, 1998; Sternberg & Grigorenko, 1997). Accordingly, this study employs cognitive styles in dealing with the adaption of the learning environment, such as the navigation modes, whereas learning styles are used to deal with the presentation modes of multi-source materials that are composed of figures, videos and texts. In this study, learning styles are used to provide personalized learning materials and presentation layouts (Liegle & Janicki, 2006), while cognitive styles are used to develop personalized user interfaces and navigation strategies (Chen, Fan, & Maredie, 2004; Chen & Macredie, 2002; Gerjets, Scheiter, Opfermann, Hesse, & Eysink, 2009).

Additionally, Paas, Tuovinen, Merriënboer, and Darabi (2005) addressed that learners' motivation had a significant relation with cognitive load, especially on mental effort. They suggested that motivation could be identified as a dimension that determines learning success, especially in complex e-learning environments (Paas, Tuovinen, Merriënboer and Darabi, 2005). The relationship between cognitive load and motivation is also stated by Moos (2009).

Chen (2008) indicated that adaptive learning systems that utilize learner preferences, learning style or learning behavior while failing to consider learner ability can result in a mismatch between learning object/content difficulty and learner ability, leading to cognitive overload and failure of the personalized learning path for the student. Chen reported on the design and initial results of an experimental learning system designed to optimize individual learning. The system used a genetic algorithm that considered courseware difficulty and concept continuity to generate an optimal learning path based on a student's incorrect responses to a course pre-test. Chen provided detailed information and graphics describing the system architecture and components, the step-by-step operational procedures, and course modeling based on Computer Assisted Testing (CAT) and Item Response Theory (IRT).

An underlying assumption was that the course content or courseware corresponding to a given test question also matched the difficulty level of the test question. Additionally, the system was dependent on SCORM meta-data maintained in the XML binding files of the content to convey course concept. Chen provided a detailed report of how the metadata was preprocessed and how the relationship of concepts between testing items and content was estimated using a vector space model. He also described the algorithm, which uses chromosomes and genes as a metaphor, to describe how the serial numbers of courseware relate to each other in order to arrive at an individualized course sequence that is mindful of the need for concept continuity in the prescribed learning path.

There are many researchers have attempted to design and develop individualized learning environments based on learning styles:

Triantafillou, Pomportsis, and Georgiadou (2002) developed AES-CS. The Witkin and Goodenough LS were employed in this system. Two different LSs, which are field dependent and field independent, were used in this system. Those who learn field dependently follow a course from general to specific while those who learn field independently follow a course from specific to general.

Arthur was designed and developed by Gilbert and Han (1999).

VAK LS model was the basis of this system, and visual-interactive, audio voiced and text-writing based content was prepared and presented to the student. The system was developed to teach C++ computer programming language.

CS383 was developed by Carver, Howard, and Lane (1999). Felder–Silverman LS was employed in this system. The system was designed for ‘‘Computer Systems’’ course.

Brown, Fisher, and Brailsford (2007) developed the DEUS system based upon the Felder-Silverman LS. The system was prepared at primary school level to teach the lifecycle and flowery plants subjects of a biology course.

eTeacher was developed by Schiaffino, Garcia, and Amandi (2008) relying upon the Felder-Silverman LS. This system was prepared in order to teach artificial intelligence course in the Department of System Engineering.

iWeaver was developed by Wolf (2003). Based on Dunn & Dunn LS, this system employed the adaptive version of this system to teach a Java programming course. The system was enriched with style based media components and other learning instruments. Based on the perceptions of individuals iWeaver presented one of four possible segments of learning content.

ILASH was developed by Bajraktarevic, Hall, and Fullick (2003). Hsiao LS was employed in this system. This system was designed to teach the characteristics of waves and solar system subjects of Physics course.

INSPIRE was developed by Grigoriadou, Papanikolaou, Kornilakis, & Magoulas (2001). Honey & Mumford LS was employed in this system.

WHURLE-LS is a system constructed on WHURLE system developed by Moore, Stewart, Zakaria, and Brailsford (2003). Based on Felder–Silverman LS, this system presented both visual and oral content to students. The system was designed and applied at Nottingham University Department of Computer Sciences and IT to teach Internet and Web 2.0 (Brown, 2007).

Mustafa and Sharif (2011) developed AEHS-LS which utilizes VARK, an acronym for visual, auditory, read/write, and kinesthetic, as the LS. This system intent was to teach JavaScript.

3. System Organization and Major Components

In terms of the system architecture, IELS consists of the following major components: an answer vector, a student model consisting of a background profile and real time profile, a lecture material depository, a knowledge base, a case base, a conflict resolution component, a fuzzy reasoning mechanism, a lecture organizer, and teachers/students GUI interfaces. The student GUI allows the student to see the lectures and receives the feedback, providing an interactive way to communicate with the system. The performance vector collects and records the answers from the student to be analyzed further. The student model allows the system assesses the student's intellectual and comprehension level. The lecture material depository stores and organizes lecture materials in a tree-shape data structure which makes it easy for the system to retrieve and organize lectures. The knowledge base provides guidance under which a set of appropriate lecture units can be accessed and retrieved. The case base stores pass experiences and make it possible for the system to select lecture materials for students with similar characteristics. The conflict resolution mechanism decides the most relevant set of lecture materials for the student. The fuzzy inference component employs fuzzy logic and reasoning to measure partial truth values of matched rules and data. It makes reasoning processes more robust and accurate. The lecture organizer with the guidance from knowledge base and case base compose a new lecture unit for students. The teacher GUI helps the domain subject expert to design lecture materials and enable the expert to enter and modify them easily.

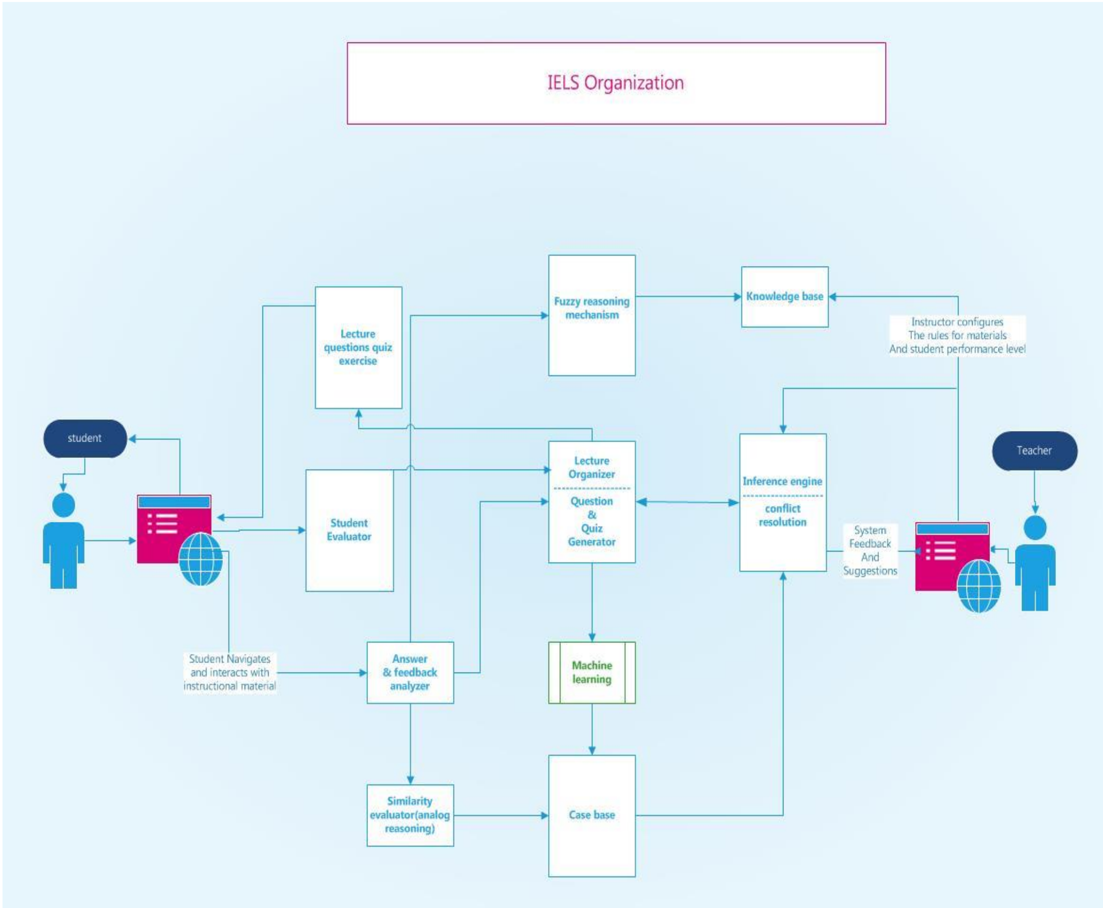


Fig. 1. IELS Business Flow Chart

In what follows, each component is discussed in details.

3.1 Student Profiles

To approach individualization in e-learning environments and to describe students' characteristics and performance, IELS builds two profiles for each student: a background profile and an active profile. The background profile contains the student's name, GPA, age, year_at_school, major, learning style preference in terms of text, video or animation, a self-assessment knowledge level in terms of advanced, intermediate or introduction of the topic being discussed. When reading course materials or studying, some students prefer to have complete silence, while others like to play music or have some other sort of background noise to stimulate their brain. IELS lets students choose the lecture format they feel comfortable with and prefer to. The age data may tell the system how mature a student is. A GPA is a good measurement on a student's aptitude and intellectual ability. It could show how well a student may master the topic and how fast a student may understand the lectures. The learning style preference helps the system to present the teaching material in a preferable format for a student. The year_at_school data shows the maturity of a student and how familiar s/he is to the learning environment. The historical data stored in the background profile enables IELS to have an initial assessment on the student and therefore provide a course content appropriate to his/her level of knowledge and aptitude at the beginning of the lecture delivery. The active profile records the actual performance of a student during the tutorial session. It captures the student's activities when s/he views tutorial materials. It includes the following data and information: the length of time viewing a segment of the tutorial, the number of examples requested for the same topic, how many rewinds if the lecture is presented in a video format, how many times of scrolling and browsing, how many clicks on a particular segment, the number of correct answers of exercises/quizzes, and the length of time spent on an exercise. Thus, a student's learning performance and pattern are then quantifiable, observable and digitalized. Generally, good students learn more quickly, deeply and broadly than their peers. They have high reasoning ability, creativity and excellent memory. A shorter time on a segment of the tutoring material and no additional examples requested in the session may indicate such a good student. To force such a student to go through a long sequence of simple explanations and examples with s/his peers is not an effective pedagogical strategy. On the other hand, below-average students may need to view the course materials repeatedly and may rewind a video segment several times and have more clicks on the segment. These digital feedbacks from the session may inform the system that the student is having difficulty in understanding the topic being discussed. The common goal for the online course delivery system is to make sure that all students can understand a given topic. The time required and the path taken to reach such a goal may be quite different depending on the student's background and intellectual ability. The traditional "one-size-fit-all" lecture needs to be improved to achieve better effectiveness and efficiency. The real-time data captured in the active profile is critical for IELS to make its decision on how to organize a lecture adaptively to a particular student. Combining the data from the background profile and active profile, IELS can guide students to appropriate instructional materials and examples. By dynamically tailoring online course materials, IELS provides individualized course content to different students based on their characteristics and comprehension.

3.2 Performance measurement Vector and Data Collection

The feedbacks from students are extremely important data for IELS to determine the student's level of comprehension and intellectual capability in order to dynamically construct a lecture unit to satisfy his/her need. For good students, the lecture content may be abstract, concise, and not too many simple examples. For students at the introductory level, the lecture content may contain detailed discussions and simple examples and exercises. To make such an assessment of a student's level of knowledge, IELS has a vector placed between the student's GUI and other components of the IELS. It collects records and measures the student's learning activity and performance during a lecture session as well as the student's statistic data. The vector consists of 14 slots, see the diagram below.

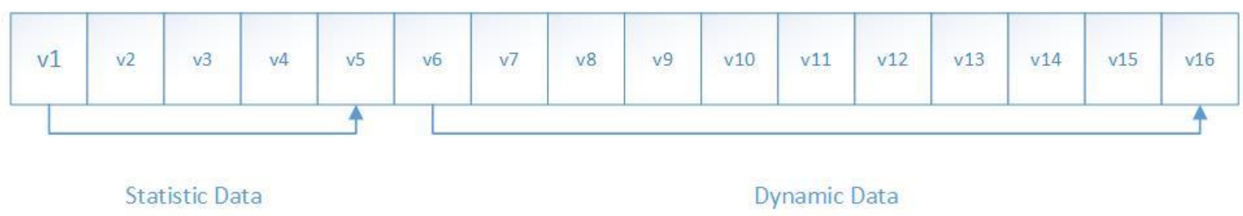


Fig. 2. Performence measurement Vector

The first 5 slots are allocated to store a student's statistic data after the student has login into the system. See below:

V1 GPA

V2 age

V3 year-at-school

V4 major

V5 self-assessment

The data collected from these slots enables IELS to make an initial assessment on the student's level of the knowledge and then accordingly prepares a set of lecture units appropriate to his/her level. IELS may, however, change its assessment and the lecture content as more real time data comes in during a lecture session. The next segment of the performance vector from V6 to V14 contains dynamic data produced by a student during his/her lecture session. It contains the following data:

V6 the length of time a student views a lecture segment

V7 the number of times a student clicks and scrolls on a page

V8 the number of times a student requests additional examples

V9 the number of correct answers to given examples/answers

V10 the number of correct answers to given exercises

V11 the number of times going between tutorial and examples

V12 the number of times going between tutorial and exercises

V13 the length of time a student views an example

V14 the number of correct answers given a set of quizzes at the end of lecture

Before, during and after each session, IELS poses questions, exercises, and quizzes to the student, and collects the feedback from the student. IELS also monitors and records the student's real time learning activity such as the number of clicks and the length of time viewing a lecture unit. These data define the level of the student, advanced, intermediate, or introduction. It helps IELS select the lecture units appropriate to the student's knowledge and comprehension relevant to the topic to be discussed in the session. The data collected during the session is used to build a student model for future reference and to construct a lecture unit for similar students that may be encountered in the future. The data collected after the session determines whether or not the student has understood the topic and decides if the student can go to a new topic. Part of the diagnostic data collected is used to match the rules in the knowledge base to dynamically select suitable lecture materials appropriate to the student's level. Part of the data is used to recall prior lesson materials recorded in the case base so that a proven and effective lecture can be provided to students with similar characteristics. Unlike most e-learning systems, IELS interacts with students and collects data to classify students into different categories based on their performance and feedbacks. With these data, IELS is able to provide personalized learning experience and to facilitate students to learn better by using different ways of selecting examples, exercises and lecture materials. The idea is to adapt both the content and presentation of the course based on a student's learning style, background, progress, and comprehension. The purpose of gathering this data is to accurately model the student, to more effectively assist the student during the lecture session and to predict how to organize the lectures for future students with similar characteristics.

3.3 Lecture Material Repository

The ability to effectively organize educational resources in terms of accessibility, reusability and interoperability lies in the construction of a repository. A lecture material repository plays crucial role in an intelligent tutoring system. It digitally stores, processes and retrieves teaching materials prepared by domain subject experts (teachers). Large amounts of content are placed in a hierarchically organized and aggregating structure on several levels. A group of lecture notes related to a particular subject resides in a tree-like data structure where the general topic is at the top and more specific sub topics are stored below. IELS employs a hierarchical structure where each node represents a concept to be learned, a set of examples associated with the concept, a set of exercises for students to self-evaluate how much they understood the concept, and a set of quizzes to test the student's comprehension of the concept being discussed. A lecture material repository is designed with four goals in mind. First, it should help teachers enter and modify teaching materials in an easy and user-friendly way. Second, it should facilitate the retrieval of relevant materials efficiently and accurately. Third, it should make the process of reorganizing a lecture unit possible and straightforward. Fourth, it should allow the system to form a new lecture with ease, flexibility and continuation. There are three kinds of nodes in this hierarchy: topics node, lecture note and module node. A topic node specifies the topic being discussed. A lecture unit node contains any number of lecture modules decided by the domain subject expert. A module node consists of tutorial, examples and exercises with different level of difficulty. See the following diagram which contains the topic nodes and lecture unit nodes.

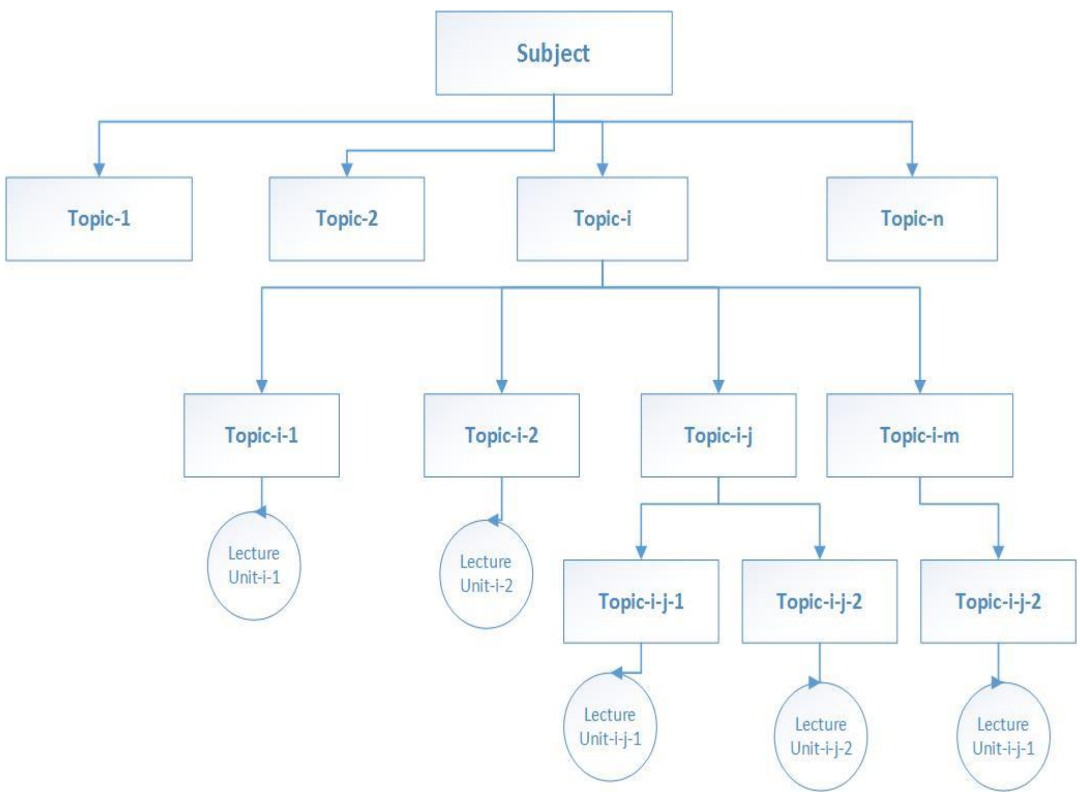


Fig. 3. Lectural Material Repository

It is worth noting the number of levels of the above hierarchy is determined by the domain subject expert. In the above diagram, for example, the Topic-2 has one level while Topic I has 3 levels. The system has flexibility to expand the structure to any depth as the domain subject expert considers necessary. At the top of the hierarchy, the node "Subject" represents a course title, such as C++ programming language, Data Structures and Algorithm Design, or Artificial Intelligence. Using the Data Structure and Algorithm Design course as an example, the nodes at the secondary level may contain topics such as Sorting Algorithms, Linked Lists, Trees, Stacks and Queues and Time Complexity Analysis. The nodes at the third level may contain sub titles within each topic domain, such as Insertion Sort, Quick sort, Merge Sort and Heap sort under the node Sorting Algorithms. More levels may be added if necessary. A lecture unit may be 50 minute, 100 minutes or 150 minutes in length. Thus, it may be further divided into smaller units called modules. See the diagram below.

Fig. 4. Lectural Unit

A lecture unit consists of one or more modules that contain the following materials:

1. Learning Object: a statement which defines the concept or topic to be learned.
2. Tutorial: explains a concept, a technique or a theory. It may be delivered as text, drawings or graphics. It also may contain multimedia and interactive content; this element may contain self verification and verification components.
3. Examples: The illustrative concrete instances of a concept are related to the course content. They serve the purpose to further illustrate the concept in a concrete way. A student may want to see more examples in order to understand a concept. With a set of examples available and associated with the concept, the student can demand and access them easily.
4. Exercises: a set of problems designed for self verification. The set of exercises help students understand the concept better, and confirm that the student's understanding of the concept is consistent with the teaching instructions and objective.
5. Assessment (quizzes): a special type of exercises which enable students to verify their comprehension and progress in a given section of lectures. IELS also relies on the answers of these quizzes to determine whether the student can start a new topic.
6. References: a list of textbooks and websites to expand on issues discussed.

One important consideration in the construction of a lecture material repository is granularity. A finer level of granularity ensures a greater potential and flexibility to create a new lecture unit but it may take too much time and effort for domain subject experts to prepare course contents. A balance must be decided to make it practical and feasible. A module is shown in the following diagram:

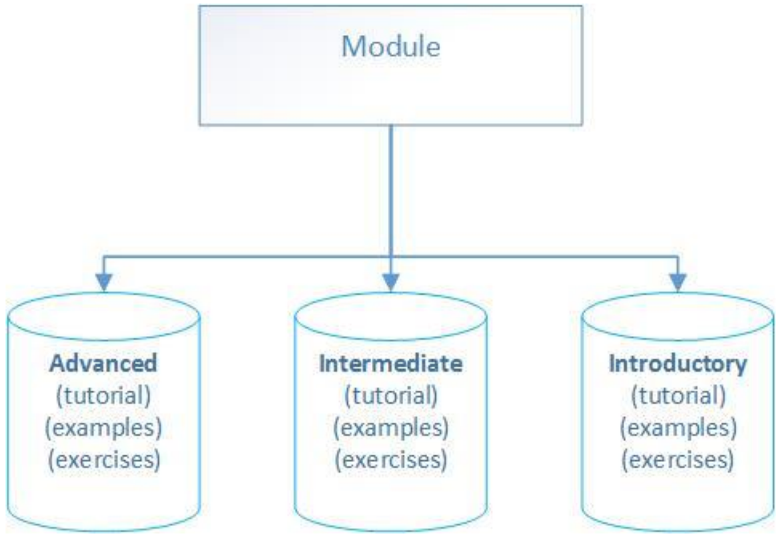


Fig. 5. Module

In our system an “atom” is defined as a module which cannot be divided further. The tutorials in a module explain a concept differently in terms of details and depth. The tutorial at the level of advance explains the concept in a concise and abstract way. It may not need to provide many examples to the student. The tutorial at the level of intermediate would provide a little more details and more examples. The tutorial at the introductory level would discuss the concept with simplest terms and elementary examples and exercises are desired. Associated with each module is a time count which contains the expected length of the module delivery. It represents the time period during which a student must realize the entire segment and is preset by the domain subject expert prior to a learning session. The modules are used as “building blocks” in the lecture reorganization process. The domain subject expert has the authority to decide how to break down a lecture and how many modules are needed to form a lecture. Designing the lecture organization in IELS as a modular concept in contrast to a course concept allows us to move from the traditional course-building approach to that of “building block concept". With these building blocks readily available, IELS is able to dynamically select modules based on the student's comprehension and progress, and adaptively form a new lecture to satisfy their needs.

A lecture unit could have one module if it is an introduction lecture, or it may have several modules if the lecture discusses a complex concept or theory. The most important reason for having the module component in the system is to allow flexibility and feasibility to dynamically deliver course content in an adaptive and continuous manner. Because each module has three levels of course content (advanced, intermediate, and introductory) to cover the same topic, IELS can choose any module of different difficult levels to form a new lecture. In other words, the modules are interchangeable if they have the same module number and are attached to the same concept node. Since it is the domain subject expert's responsibility to break down the lecture into modules, the continuation of a lecture is guaranteed. See the following diagram of 3 students with different combinations of modules:

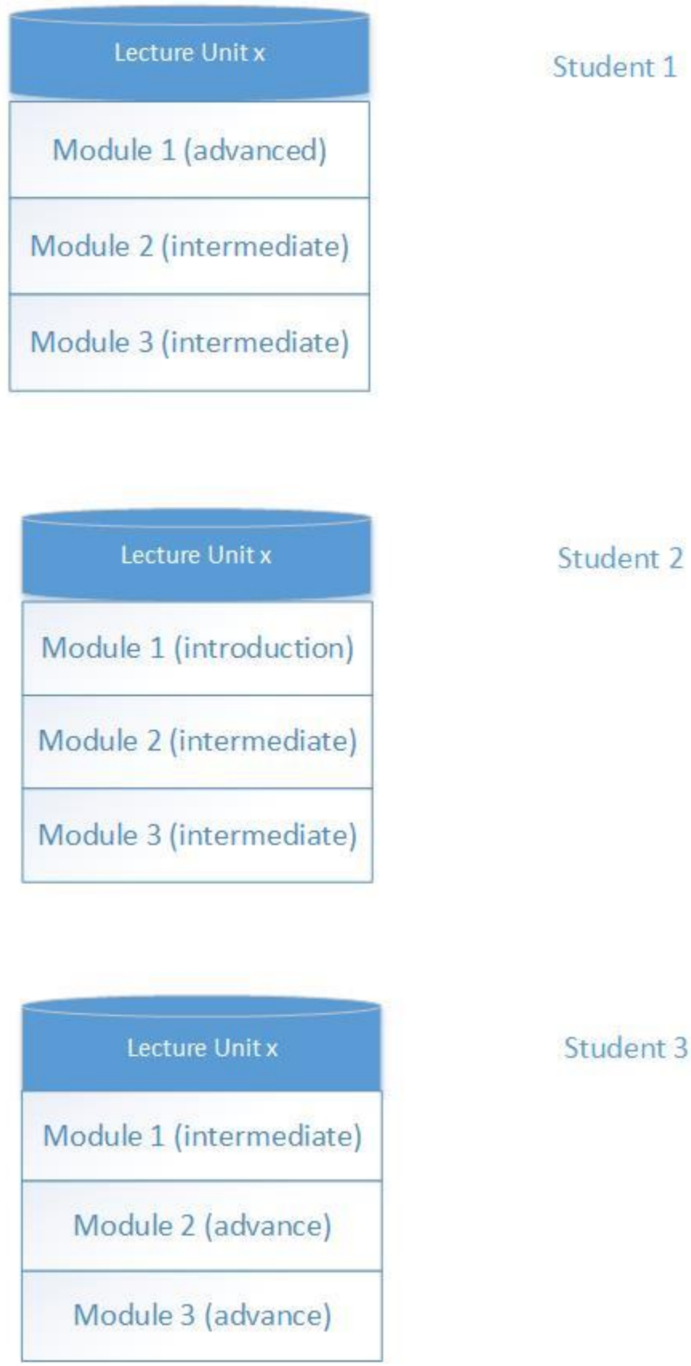


Fig. 6. Lecture Units

In the above diagram for the same lecture topic x, three students are given three different combinations of lecture materials. Student 1 started with an advanced segment but had difficulty in following the lecture. He then was given the course content at the intermediate level. Student 2 started with an introductory content. The real time performance indicated he mastered the topic very quickly and then was given the teaching materials at the advanced level. Student 3 progressed from the intermediate level to the advanced level in the same session. To adaptively present course contents to students based on their comprehension and competence implements an effective educational strategy in e-learning systems. By providing lecture content at different levels to students, IELS makes it possible for students to learn new concepts at their own speed and level of comprehension. Good students can finish the lecture much faster than others, while the students at the introductory level may take longer. With detailed explanations and more examples, they, hopefully, would eventually achieve the same lecture objectives. This approach implements an effective pedagogical strategy not seen in traditional online course content delivery systems.

4. Theoretical Framework and feature

4.1 Knowledge Base

The knowledge base captures the knowledge of domain subject experts in the form of If-Then rules. It gets its power from domain subject experts that have been coded into facts, rules, heuristics and procedures. This knowledge of a teacher is stored in the knowledge base separate from other control and inference components in IELS. It is possible to add new educational strategies and methods in the system easily. It is also easy to modify the existing knowledge already in the knowledge base. Researchers believe that a single rule corresponds to a unit of human knowledge. In the design of an intelligent tutoring system, a knowledge representation for human problem solving expertise is a critical and complex task. The main purpose of the rules in the knowledge base is to determine a student's comprehension and intellectual level so that appropriate teaching materials can be retrieved from the Lecture Depository. In what follows, the knowledge representation in IELS is described along with actual rules as examples.

RULE <ID>

IF {(<condition_1 RO value_1 >)

LO (<condition_2 RO value_2>) …..

LO (<condition_n RO value_n>)}

THEN (Recommendation)

UNLESS (statement)

REASONS (statement_1, statement_2… statement_m)

WITH (CERTAINTY FACTOR = <real number>)

Where the term LO represents a logical operator, such as AND, OR and NOT. The term RO represents one of the following relational operators: equal to (==), less than and equal to(<=), not equal(!=) to and greater than and equal to(>=). These operators together with the conditions in the Condition clause express a certain requirement that must be satisfied before the rule can make a recommendation listed in the THEN clause. The UNLESS specifies an exception that prohibits the rule to be triggered and fired. The REASONS clause lists the reasons in order to explain and justify why a conclusion is reached. The WITH clause contains a real number in the range of (0...1) measuring the strength of evidence in support of the rule's hypothesis. It expresses how much confidence we should trust this rule and its recommendation. There are two kinds of knowledge captured in the form of IF-Then in IELS: an initial assessment on students' levels before a lecture and a final assessment on students' performance in the lecture. The assessment data for diagnostics comes from the student's background profile. It provides an initial assessment on the level of a student's knowledge, intellectual ability and comprehension so that IELS can provide individualized teaching materials appropriate to students to begin a lecture session. If this decision is not accurate, IELS is capable of making adjustments during the lecture session. More detailed discussions are presented later in the dissertation.

Three actual rules in the knowledge base are listed below:

RULE <5>

IF ("year_at_school" == sophomore)

AND ("GPA" == excellent)

AND ("major" == topic)

AND ("self_assess" == advance)

THEN (Student_Level is advance)

REASONS (good intellectual ability and motivated, mature, familiar with the topic)

With (CERTAINTY FACTOR = 0.95)

RULE <12>

IF ("year_at_school" == freshman)

AND ("GPA" >= average)

AND ("major"! = topic)

AND ("self_assess" >= intermediate)

THEN (Student_Level is introduction)

UNLESS (age < 17)

REASONS (student is new to university, not familiar with the topic. With exceptions of prodigy)

With (CERTAINTY FACTOR = 0.8)

RULE <13>

IF (“year_at_school” == freshman)

AND (“age” == old)

AND (“major” == topic)

AND (“self_assess” >= advanced)

THEN (Student_Level is advanced)

REASONS (student is new to university, adult student. May have years of working experience in the field)

With (CERTAINTY FACTOR = 0.74)

These rules are self explanatory. With the recommendations from the rules in the knowledge base, IELS has initial assessments to select lecture materials including tutorial, examples, exercises and quizzes appropriate to the student’s ability and comprehension. The important building blocks of a lecture consist of tutorials, examples, exercised and quizzes. By selecting different building blocks, IELS is able to provide individualized course content to students of different levels. For instance, an advanced student could skip some examples to speed up his/her session, while a student at the introductory level may need to see more examples and to spend more time on lecture materials in order to understand a new concept.

Another kind of knowledge represented by the IF-THEN rules in the knowledge base is used to evaluate the student's performance during a lecture session. For the same topic, a lecture unit may consist of different tutorials, different examples and different exercises. Good students may finish the lecture much faster than others, while the students at the introductory level may spend more time on the same topics. At the end all students are expected to meet the same learning objective. Before students can continue to next new topic there is a same set of quizzes/questions at the end of the lecture to test the students' understanding. Some of the rules in the knowledge base are presented below:

RULE <17>

IF (TMS >= very_long)

AND (NC >= too_many)

AND (NE >= avg)

AND (CAQ = avg)

AND (CEXE <= avg)

THEN (Student_Level is current_level -2)

REASONS (took too much time to study, requesting too many examples, fewer correct Answers to quizzes and exercises)

With (CERTAINTY FACTOR = 0.8)

Where TMS is the time spent to view tutorial; NC is the number of times the student clicked and scrolled during a session; NE is the number of times the student requested additional examples; CAQ is the number of correct answers to the quizzes and CEXE is the number of correct answers to exercises. It seems that the student in this example showed some difficulty in the learning session even though s/he managed to answer half of the quizzes correctly. The system decides to provide a little easier course content for the next course content. If the level of difficult cannot be lowered, the student needs to retake this lecture unit.

RULE <18>

IF (TMS >= short)

AND (NC == fewer)

AND (NE >= 0)

AND (CAQ = excellent)

AND (CEXE <= excellent)

THEN (Student_Level is current_level +1 if possible)

REASONS (showed no problems to handle teaching materials, no additional examples needed. Answered the quizzes correctly)

WITH (Certainty Factor = 0.9)

If a student's learning pattern and activity can match Rule 18, it shows that the student can master the course content with ease and a more advanced content should be used in the next lecture session if possible.

One important implementation issue here is how to convert a number to a literal term. For example, how a recorded number of clicks and scrolls can be converted to literal terms such as "too-many", "fewer" or "average"? Or an actual age in the form of a number should be converted to terms like "too young" or "too old"? This is the task that can be handled by a fuzzy reasoning component that converts numerical data to linguistic terms. In the fuzzy reason process partial set memberships are allowed. A membership value may be any number between 0 and 1. A 0 represents the concept of "completely false" and a 1 represents the concept of "completely true". Anything is between represents linguistic and imprecise terms like "advanced", "intermediate" and "introduction". A rule in IELS may be triggered by a partial truth value. That is, a student may be considered 70% advanced and 30% intermediate. More details will be discussed in the section of Fuzzy Logic and Reasoning later in the dissertation. One important consideration in the design of knowledge bases is how to enter, modify and verify the rules in the knowledge base. IELS provides a user-friendly GUI interface for domain subject experts to input their knowledge and experience into the system. It has a template-based editor which consists of slots similar to the rule described above. Except for the rule ID numbers, the teacher can easily enter his/her lectures including examples, exercises and quizzes. By specifying a subject, a topic or a segment, IELS automatically stores a lecture unit in an appropriate place in the Lecture Material Depository.

4.2 Case Base and Analogical Reasoning

Analogical reasoning and learning have grown to become a central research subject in artificial intelligence and intelligent tutoring systems in the recent years. Researchers believe that analogical problem solving provides a promising approach for the acquisition and effective use of knowledge. The major influence of cognitive science on case-based reasoning (CBR) is centered on fundamental concepts such as experience, memory and analogy. CBR addresses the ways experiences are recalled to be used in reasoning, the use of old experiences in reasoning and the ways in which new experiences are analyzed, indexed, and stored in memory. The computational model of CBR implements these concepts in the process of reasoning and enables an intelligent system to recognize similar situations, recall past useful and related experience, stores successful experiences and adapts for new but similar problems encountered in the future. This model also provides an explanation of the role memory plays in reasoning -- how memory is accessed during reasoning and how reasoning contributes to changes in the content and organization of memory. The CBR stores a case that consists of a problem description part and a solution part. The description part has as a list of attributes describing the characteristics of the case. The solution part contains a collection of plans, advices or recommendations that work well for the problem. The CBR process consists of 4 phases: recognition, retrieval, reuse, and storage. In the recognition phase, the system compares the new situation with many cases stored in the case base. A similarity score is calculated to measure how similar the current situation is with previous scenarios. In the retrieval phase one or more relevant cases are retrieved from the case base based on the similarity score. The implication here is that past experience may provide guidance and clues for the system to solve the current problem more effectively. In the reuse phase, one or more recalled cases from the case base are applied to the current situation. There are two possible ways to proceed: 1) directly apply the solution in the solution part to the current situation if it is considered similar enough, and 2) the solution may need to be modified in such a way that it would be more appropriate to the current problem. In the storage phase, a new case is stored in the case base as a new addition. Thereby, the new problem solving experience becomes available for future episodes. Over time, a case base would grow as more and more problems are solved and analyzed. The “IQ” of the intelligent system will grow over time. It implements the concept of learning in an intelligent system since the problem solving ability of an intelligent system will improve as it encounters more and more problems. One important issue is to organize and store cases in a knowledge structure that future search and retrieval may be conducted in an efficient way.

Bloom et al proposed three aspects of educational activities: cognitive, affective and psychomotor. The most important activity in the field of online course delivery is cognitive. It may be classified into the following categories: knowledge, comprehension, application, analysis, synthesis, and evaluation. An example in describing a complex concept is an intuitive way to help a student acquire knowledge, understand abstract ideas, apply instances to new concepts, and analyze what has been learned. However, an example may work very well for a group of student but may not make much difference to others. Depending on the deficiency in knowledge and intellectual ability, we need to select appropriate examples for students with different characteristics. During a lecture, some students may become confused by some new topics in the lecture. They need to see specific instances, concrete clues, and particular hints. Appropriateness of examples is closely related to the effectiveness of course content delivery.

The case base in IELS maintains a rich set of cases (scenario) of student's learning patterns, their performance and backgrounds along with a set of examples, exercises and quizzes appropriate to their comprehension. A case consists of an attribute part and a recommendation part. When comparing a new case to the cases in the case base, the values in the attribute part are calculated. A similar score indicates how similar they are. Once a case is considered similar enough, the content in its recommendation part is retrieved and applied to the new case. The content in the recommendation part usually contains a combination of modules along with a set of examples and exercises. Its components are shown below:

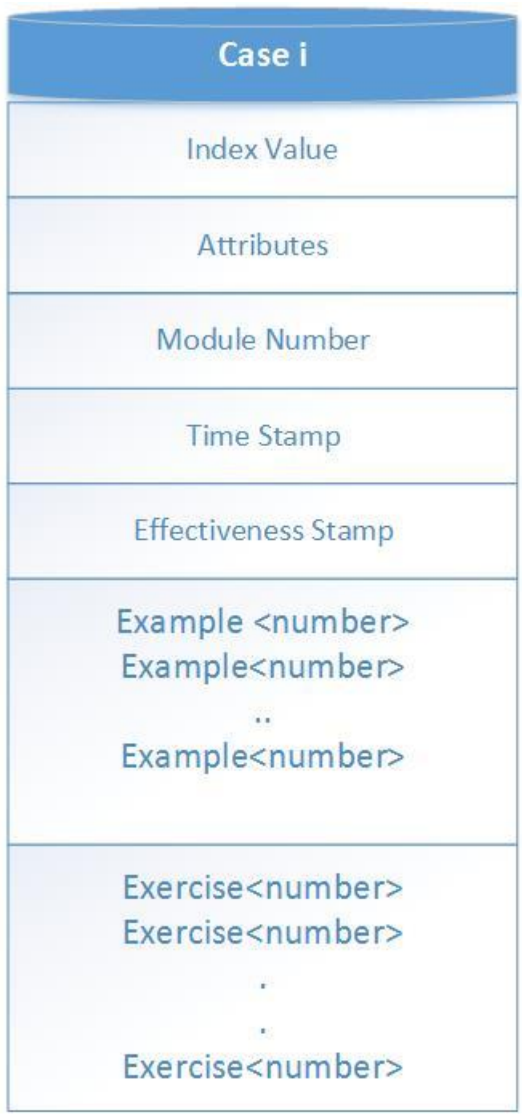


Fig. 7. Case and its Components

Case and its Components

Each case in the case base has a time stamp attached to it. Every time a case is retrieved, the value in the time stamp is increased to indicate its appropriateness. It also has an effectiveness slot. If the retrieved case can be applied to the current student successful, the value in the effectiveness slot is increased to indicate its examples and exercise are very helpful in the process of learning. If some cases are stored there for a long time without being retrieved and applied, it will be deleted eventually. This process demonstrates a form of machine learning in the design of IELS. Each case in the case base of IELS is a unique teaching strategy: given a topic and a particular group of students with common characteristics, what are the best set of examples and exercises to explain the concept most effectively? The index value in a case represents a path and a module number where the case should be attached to the Lecture Repository Hierarchy. It begins with the subject, followed by a topic path and a module number. The recommendation part consists of a list of examples and exercises. Each example or exercise has a number corresponding to the location where is should be inserted into the tutorial segment. The attributes in each case are used to measure a student's dynamic activity in a session of lecture: the time viewing a lecture segment, the number of clicks on a page of lecture content, the number of examples requested, the number of correct answers to a set of exercises, the number of correct numbers to a set of quizzes, the number of times going back-and-forth between a tutorial segment and an example, the number of times going back-and-forth between an exercise and a tutorial segment. The abbreviations and their meanings are presented in the following table.

TMS	The time length viewing a segment
NC	The numbers of times clicking and scrolling
NE	The number of times requesting additional examples
CEXA	The number of correct answers to examples
CEXE	The number of correct answers to exercises
BTE	The number of times between Tutorial and Examples
BTX	The number of times between Tutorial and Exercises
TME	The time length viewing an example

Table1. Table of attributes of a case

The attribute part of a case is also known as features. In the following formula the 8 features represent the 8 abbreviations listed in the above table.

Let N be a case with 8 features:

N = {n1 n2. n8}

And O is an old case with 8 features:

O = {o1 o2 ... o8}

CF denotes a common feature set

CF = {c1 c2 ... ck} where 1 <= k <= 8

Where { C1 and O1 = TMS

C2 and O2 = NC

C3 and O3 = NE

C4 and O4 = CEXA

C5 and O5 = CEXE

C6 and O6 = BTE

C7 and O7 = BTX

C8 and O8 = TME }

Since some attributes may be missing, the number of common features k may be fewer than 8.

Thus, a similarity score, S (N, O), of a new case N with respect to an old case O is given as:

$$S(N, O) = \frac{\left[\sum_{i=1}^{k} \lambda i \times c i\right] \times k}{8} \qquad 1 \leq i \leq 8$$

Where λi is a weight assigned to the ith feature of a case.

A high similarity score indicates the current student is similar to some past students in terms of real time learning activity and performance. By providing a set of examples and exercises proved to be effective in the past, the current student can shorten the learning curve and master the new concepts quickly.

Several cases may have the same set of modules with the same difficulty level, but the example set and exercise set may be different. Let us assume that the student is working on the Module 1. Based on his real time activity, the system matches two cases in the case base that have been used by other students in the past, see the following diagram. Even though these two cases recommend the same modules for organizing the following lecture unit, they have different sets of examples and exercises. Some examples and exercises are better than others in answering the students' questions and in helping student's understanding of a topic. Over time IELS is able to select the best combination of examples and exercises appropriate to the students with similar characteristics and therefore, to provide an effective lecture content delivery through its past experience.

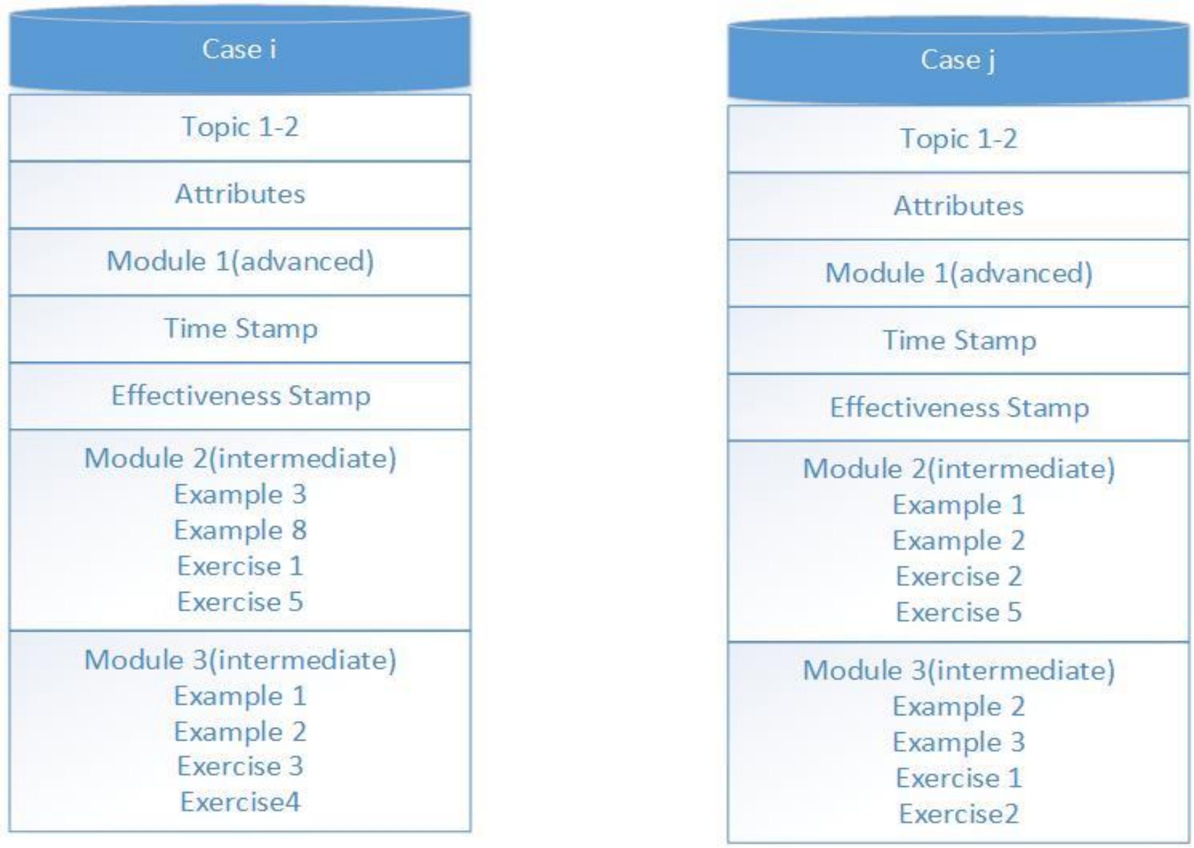


Fig. 8. Case and its Components examples

The cases presented above show the same module recommendation but with different sets of examples and exercises. Initially IELS cannot tell which recommendation is the best fit to the students' need. By using the Effectiveness stamp, the most effective combination is identified eventually and will be used in the future course content delivery. Once the recommendation from a case is selected, it is then sent to the Lecture Organizer to construct a new lecture. It is worth pointing out that the initial set of cases is created by students in the early stage of the test and development. IELS simply records and stores what examples and exercise student choose during a learning session in the case base. Some of the cases can even be provided by the domain subject expert. By the process of "trial by errors", IELS will figure out the best combination of examples and exercises for a group of students with similar learning performance and deficiency.

4.3 Fuzzy Reasoning

English abounds with vague and imprecise concepts, such as "The age of that student is old" or "The difficulty level of this lecture is advanced" Such statements are difficult to translate into more precise language without losing some of their semantic value. Let us assume the average GPA of college students is 2.5. The GPA of 3.8 is considered high and the GPA of 1.6 is low. How would a GPA of 2.9 be ranked? The use of fuzzy logic allows for more gradual changes between categories and allows for a representation of certainty in the rule consequence through the ability to fire rules with varying strength dependent on the antecedents. One of the major tasks in the design of IELS is to codify the decision-making process. Exact reasoning strategies that make use of standard probability theory. A common approach is the certainty factor (CF) in rule-based systems. CF has a value between -1 and +1, representing 100% false and 100% true respectively. Degrees of truth are often confused with probabilities. However, they are conceptually distinct; fuzzy truth represents membership in vaguely defined sets, not the likelihood of some event or condition. Approximate reasoning is needed when the assumptions necessary to apply a probability based approach cannot be met. Fuzzy reasoning uses a collection of fuzzy membership functions and rules (instead of Boolean logic) to reason about data. There is an important distinction between fuzzy logic and probability. Both operate over the same numeric range, and at first glance both have similar values: 0.0 representing False (or non-membership), and 1.0 representing True (or membership). However, there is a distinction to be made between the two statements:

The probabilistic approach yields the natural-language statement, "There is an 80% chance that student would make an A in this semester" while the fuzzy terminology corresponds to "The student's degree of membership within the set of excellence is 0.80." The semantic difference is significant: the first view supposes that the student is good or not; it is just that we only have an 80% chance of knowing it. By contrast, fuzzy terminology supposes that the stock is "more or less" doing very well, or some other term corresponding to the value of 0.80. Further distinctions arising out of the operations will be noted below.

For independent events, the probabilistic operation for AND is multiplication, which (it can be argued) is counterintuitive for fuzzy systems. For example, let us presume that x is a company, S is the fuzzy set of high P/E companies, and T is the fuzzy set of investor-preferred companies. Then, if S(x) = 0.90 and T(x) = 0.90, the probabilistic result would be:

$$S(x) * T(x) = 0.81$$

Whereas the fuzzy result would be:

$$\mathrm{MIN}\{S(x), T(x)\} = 0.90$$

The probabilistic calculation yields a result that is lower than either of the two initial values, which when viewed as "the chance of knowing" makes good sense.

However, in fuzzy terms the two membership functions would read something like "x is a good student" and "x works hard" If we presume for the sake of argument that "very" is a stronger term than "fairly," and that we would correlate "fairly" with the value 0.81, then the semantic difference becomes obvious. The probabilistic calculation would yield the statement

If x is a good student and x works hard, then x is good and works hard with 0.81 certainties.

The fuzzy calculation, however, would yield

If x is a good student and x works hard, then x is good and works hard with 0.91 certainties.

Another problem arises as we incorporate more factors into our equations (such as the fuzzy set of actively-traded companies, etc.). We find that the ultimate result of a series of AND's approaches 0.0, even if all factors are initially high. Fuzzy theorists argue that this is wrong: that five factors of the value 0.90 (let us say, "Very") AND's together, should yield a value of 0.90 (again, "very"), not 0.59 (perhaps equivalent to "somewhat").

Similarly, the probabilistic version of A OR B is (A+B - A*B), which approaches 1.0 as additional factors are considered. Fuzzy theorists argue that a sting of low membership grades should not produce a high membership grade. Instead, the limit of the resultant membership grade should be the strongest membership value in the collection.

Another important feature of fuzzy systems is the ability to define "hedges," or modifier of fuzzy values. These operations are provided in an effort to maintain close ties to natural language, and to allow for the generation of fuzzy statements through mathematical calculations. As such, the initial definition of hedges and operations upon them will be a subjective process and may vary from one statement to another. Nonetheless, the system ultimately derived operates with the same formality as classic logic. For example, let us assume x is a company. To transform the statement “x is a good student in terms of his GPA” to the statement “x is a very good student in terms of his GPA. The hedge “very” can be defined as follows:

"Very "A(x) = A(x) ^2

Thus, if good(x) = 0.8, then Very good(x) = 0.64. Similarly, the word "more or less" can be defined as Sqrt(Expensive(x)). Other common hedges such as "somewhat," "rather," and "sort of," can be done in a similar way. Again, their definition is entirely subjective, but their operation is consistent: they serve to transform membership/truth values in a systematic manner according to standard mathematical functions. From the above discussion, it is clear that fuzzy logic can describe the investor's decision making process in a more natural and accurate way than the probability theory.

4.3.1 Fuzzy linguistic variables and membership

Fuzzy logic allows for set membership values to range (inclusively) between 0 and 1, and anything in between representing linguistic and imprecise terms like "slightly", "quite" and "very". Specifically, it allows partial membership in a set. It is related to fuzzy sets and possibility theory. Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is robust and approximate rather than brittle and exact. In contrast with "crisp logic", where binary sets are either true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions.

The basic fuzzy reasoning process in IELS is summarized as follows:

- Under FUZZIFICATION, Convert numeric data to literate words using fuzzy membership functions, and determine the degree of truth for the word. It calculates the degree to which the input data match the condition of the fuzzy rules.

- Under INFERENCE, the truth value for the condition of each rule is computed using 'AND', 'NOT' or 'OR,' and applied to the conclusion part of each rule. The result is one fuzzy subset to be assigned to the output variable for each rule. The output of each rule is scaled by the rule condition's computed degree of truth.
- Under COMPOSITION, all of the fuzzy subsets assigned to the output variable are combined together to form a single fuzzy. The operation SUM takes the point wise sum over all of the fuzzy subsets.
- DEFUZZIFICATION: convert the fuzzy output set to a numeric value. IELS uses the MAXIMUM method. It selects the maximum value of the fuzzy sets as the crisp value for the output variable.

Let us assume we need to convert a specific GPA value into linguistic terms such as very low, low, average, high and very high. We define the following fuzzy member function:

$$\text{GPAverylow}(x) = \begin{cases} \frac{1.5-x}{1.5} & 0 < x < 1.5 \\ 0 & x \geq 1.5 \end{cases}$$

$$\text{GPAlow}(x) = \begin{cases} \frac{x}{1.5} & 0 < x < 1.5 \\ 2.5 - x & 1 < x < 2.5 \\ 0 & x \geq 2.5 \end{cases}$$

$$\text{GPAavg}(x) = \begin{cases} 0 & x \leq 1.5 \\ x - 1.5 & 1.5 < x \leq 2.5 \\ 3.5 - x & 2.5 < x < 3.5 \\ 0 & x \geq 3.5 \end{cases}$$

$$\text{GPAhigh}(x) = \begin{cases} 0 & x \leq 2.5 \\ x - 2.5 & 2.5 < x \leq 3.5 \\ \frac{4-x}{0.5} & 3.5 < x < 4 \\ 0 & x \geq 4 \end{cases}$$

$$\text{GPAveryhign}(x) = \begin{cases} 0 & x \leq 3.5 \\ \frac{x-3.5}{0.5} & 3.5 < x < 4 \\ 1 & x \geq 4 \end{cases}$$

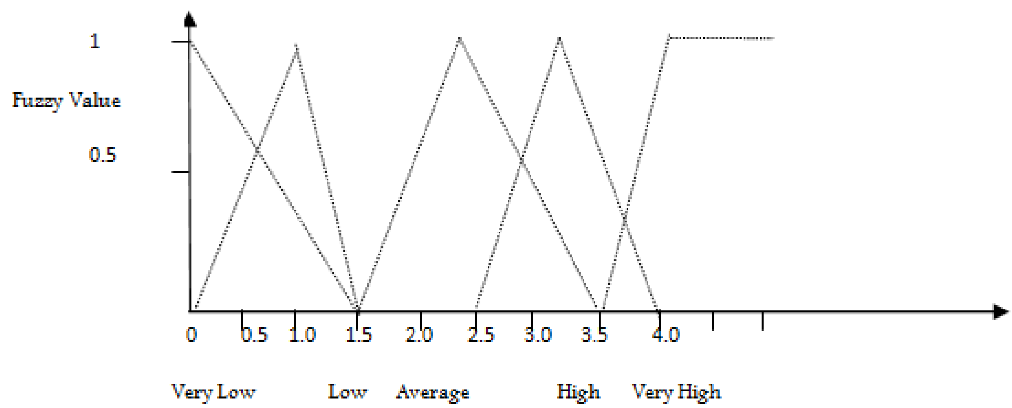


Fig. 9. Fuzzy Value

For example, a GPA of 3.8 is considered 20% of high and 80% of very high. That is, this value belongs to two member sets: high and very high with different certainties. With the tool like fuzzy reasoning, IELS is able to handle any specific number and to convert it into vague and imprecise linguistic term. It is a very critical step since many rules in the knowledge base and cases in the case base are often expressed in vague and imprecise language.

4.4 Lecture Organizer and Conflict Resolution

The lecture organizer component in IELS is a command center responsible for constructing a new lecture unit. It receives recommendation from the knowledge base and the case base. The knowledge base analyzes the student's intellectual ability and background and makes a judgment on the difficulty level of course content. The main factors to be considered include the student self assessment, maturity, familiarity to the subject being discussed, and his/her past academic performance. The case base suggests a set of examples and exercises which have be proven to be effective for students with similar characteristics. With the employment of fuzzy reasoning, it is possible that more than one rule has been triggered and fired. IELS relies on the certainty factor attached to each rule to decide which rule can be trusted. The fuzzy value that matches the rule is also considered to determine its usefulness in the decision making process. When receiving a recommendation from a selected case, IELS examines the effectiveness value. The high value is the more effective it was in the past. The value in the time stamp tells the system how often this case is recalled and applied. Combing the above factors and values, IELS select the most relevant and effective course content along with a set of examples and exercises for the current student. In the case where there is no closely matched case found in the case base, the lecture organizer component relies on the recommendation from the knowledge base to form a new lecture content. As discussed earlier, a domain subject expert decides how to divide a lecture unit into modules, and how many modules are needed to cover the topic. One module of any difficulty level can be combined with any other module as long as the module numbers are in sequence. In other words, the end of the module n -1 lecture is always followed by the first tutorial segment in module n. Let us assume that a student is working on the module 1, any module 2 with a difficulty level of advanced, intermediate, or introductory can be selected to provide a continuous lecture. With these modules as building blocks, the Lecture Organizer component can easily construct a new lecture.

5. User Interface and Flow Chart

5.1 IELS Development and Flow Chart

The IELS Built was built using Microsoft technologies. Specifically:

- ASP.NET using C# programming language is a powerful platform for building dynamic web applications that provides a tremendous amount of flexibility and power for building just about any kind of web application from small, personal websites through to large, enterprise-class web application.
- Entity Framework also known as EF is a set of open source technologies for .NET framework. Often the underlying technologies are referred as ORM (Object-Relational Mapping). It gives a higher level of abstraction to the developers to work with data with less code and without concerning the actual design of the physical database.
- Microsoft SQL Server 2012 is a relational database management system (RDBMS) designed for the enterprise environment. Like its predecessors, SQL Server 2012 comprises a set of programming extensions to enhance the Structured Query Language (SQL), a standard interactive and programming language for process information from database

- Model View Controller known as MVC is a pattern, in computer science, for developing software that allows developers to create abstractions by keeping different layers completely separate.

The intelligent and effective e-learning system includes user interfaces to both manage users and content. There are three different actors in this system: the student, the domain subject experts, and the super admin.

The super admin has permission to delete, edit, and add all system user accounts. The domain subject matter experts add, edit, and otherwise administer all course or domain content including lectures, videos, instructional materials exercises, quizzes, tests, and other feedback mechanisms. The student accesses and reviews the materials through their system interface, performs exercises, and takes quizzes.

The intelligent and effective e-learning system (IELS) records the student's performance in exercises and quizzes and determines the lecture module's comprehension difficulty level through a knowledge base and case-based reasoning. The flow chart shows if the student needs to register before starting a lecture. If the student has saved his progress in the same lecture, then the lecture organizer will direct him to the remaining sections. The static module collects data from the diagnostic test, and the dynamic module collects data during the learning session.

Since the knowledge base captures the knowledge of domain subject experts in the form of If-Then rules, the domain subject experts play a critical role in IELS. Domain subject experts not only construct If-Then rules for IELS but also the structure of the teaching material repository. Additionally, they will prepare some of the teaching material. In the flow chart, the rule engine designer can be the domain subject expert.

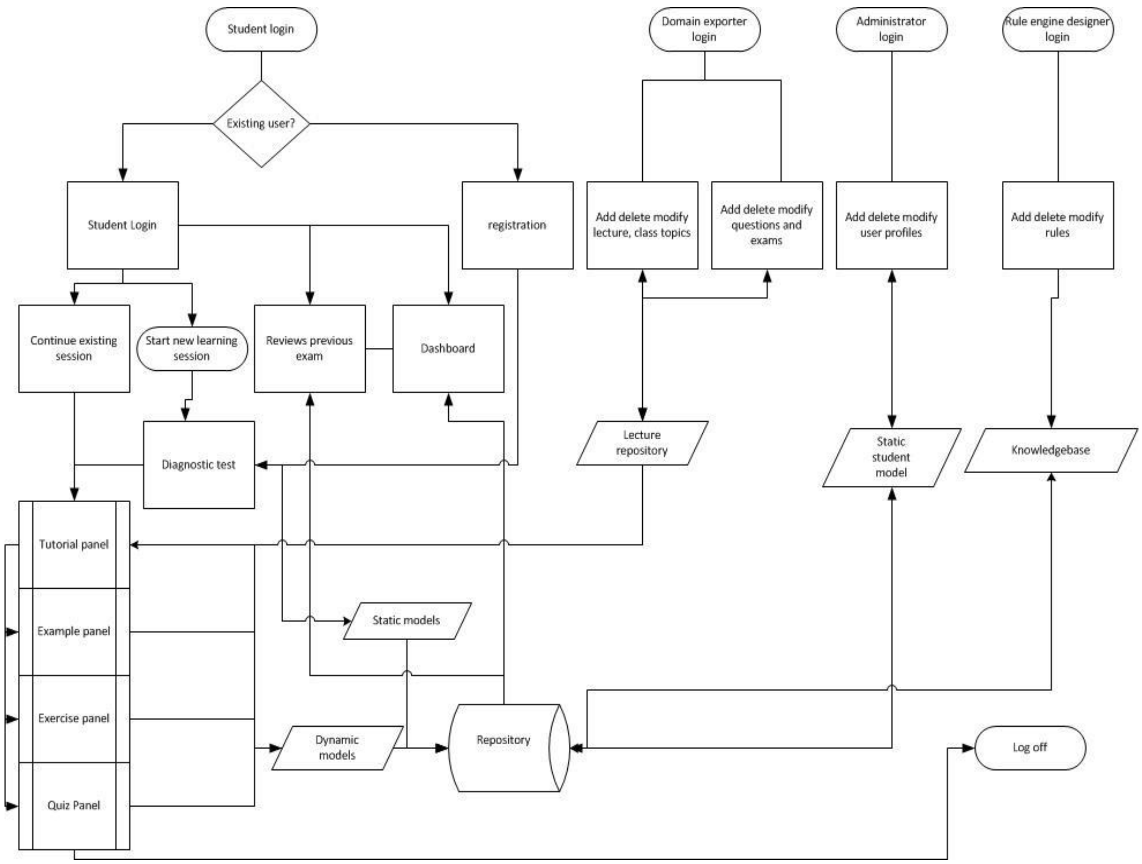


Fig. 10. Flow chart of IELS

5.2 Domain Expert GUI

The Graphic User Interface (GUI) helps the domain subject expert design lecture materials and enables the expert to enter and modify them easily. Figures 11—18 is presenting screen shots of the domain subject expert login.

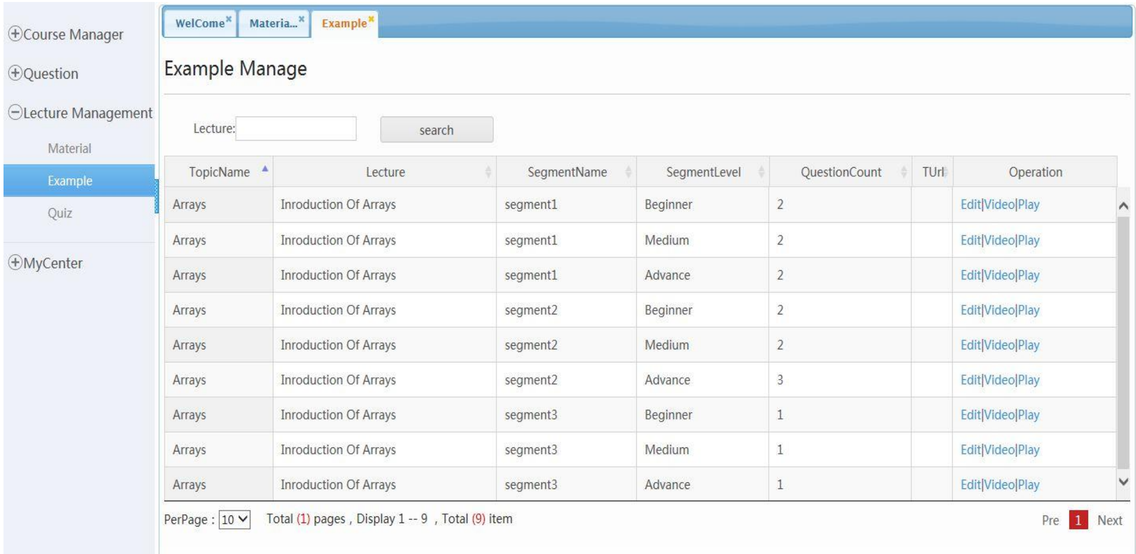


Fig. 11. Example Management Screen

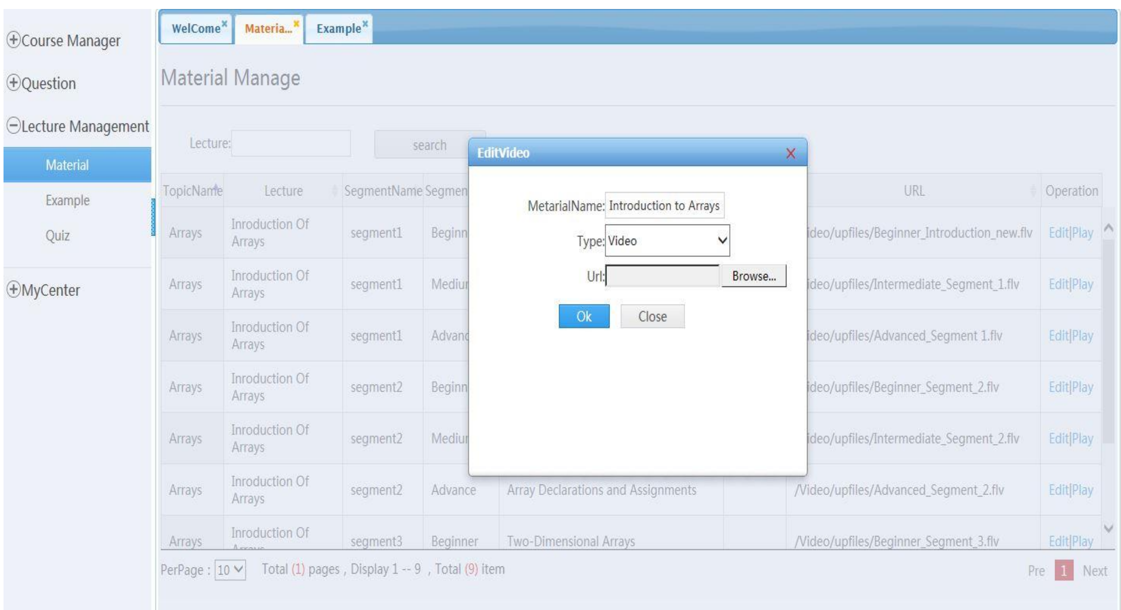


Fig. 12. Teaching Material Management

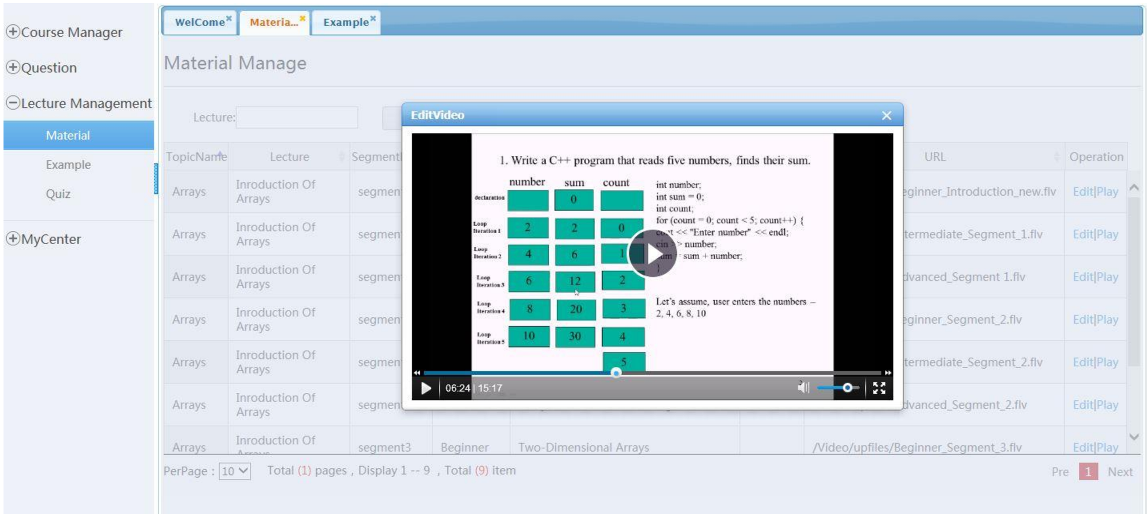


Fig. 13. Lecture management Screen

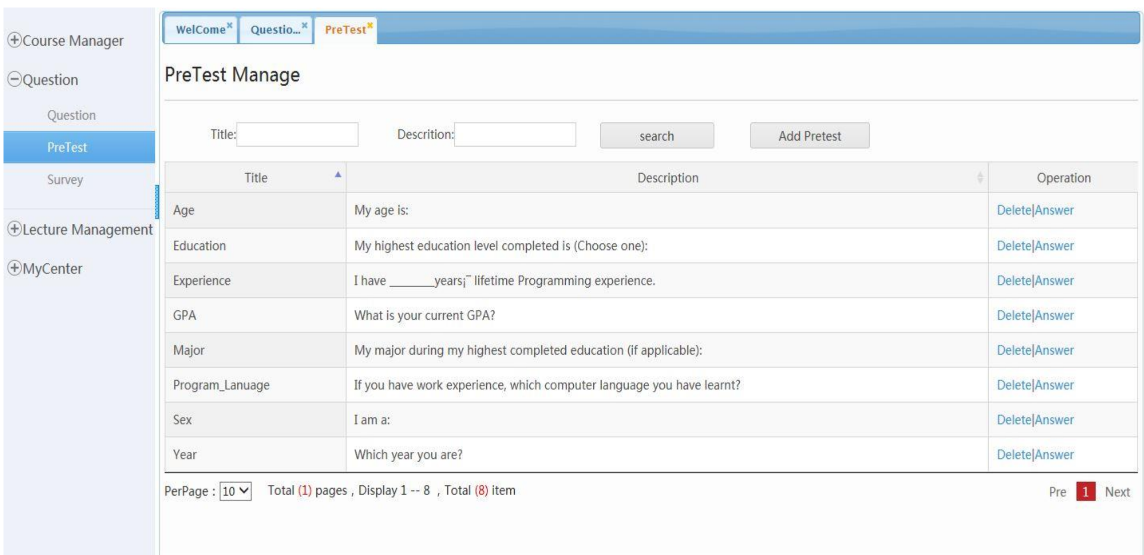


Fig. 14. Pre_Test Management Screen

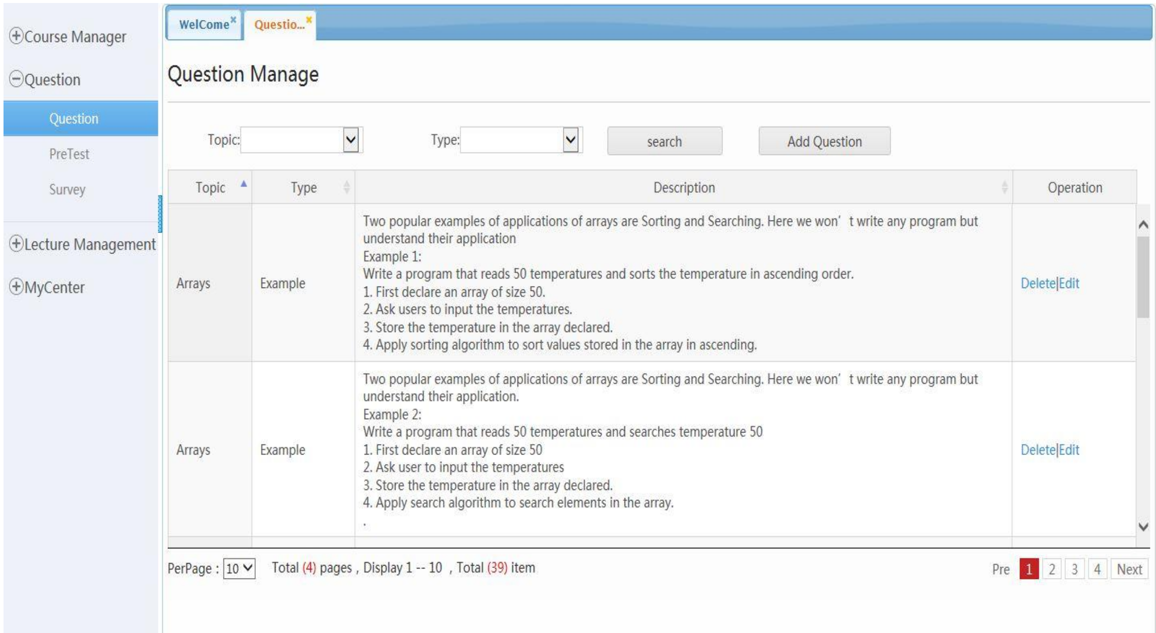


Fig. 15. Question Management Screen

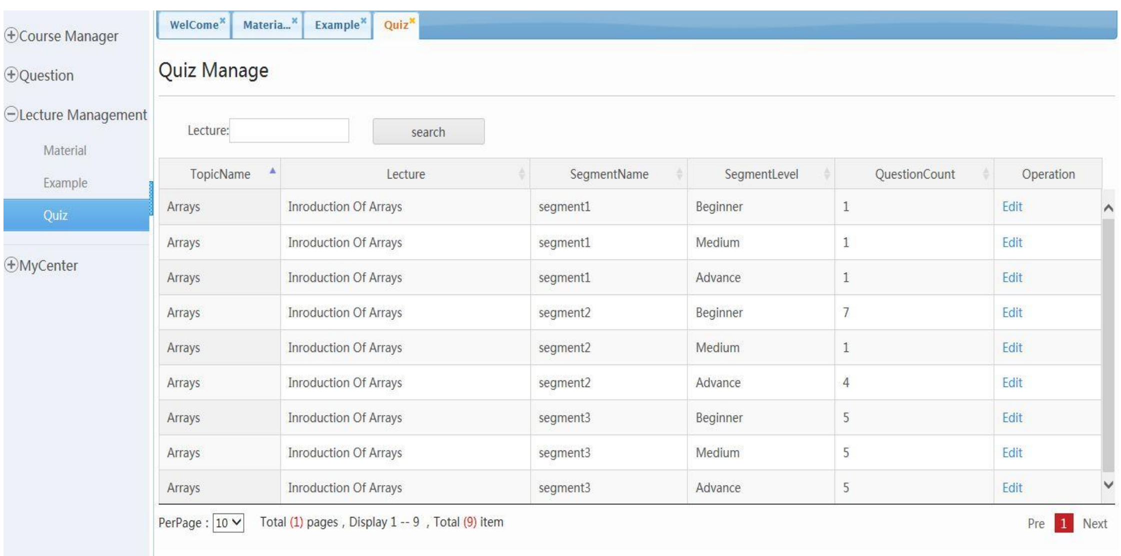


Fig. 16. Quize Management Screen

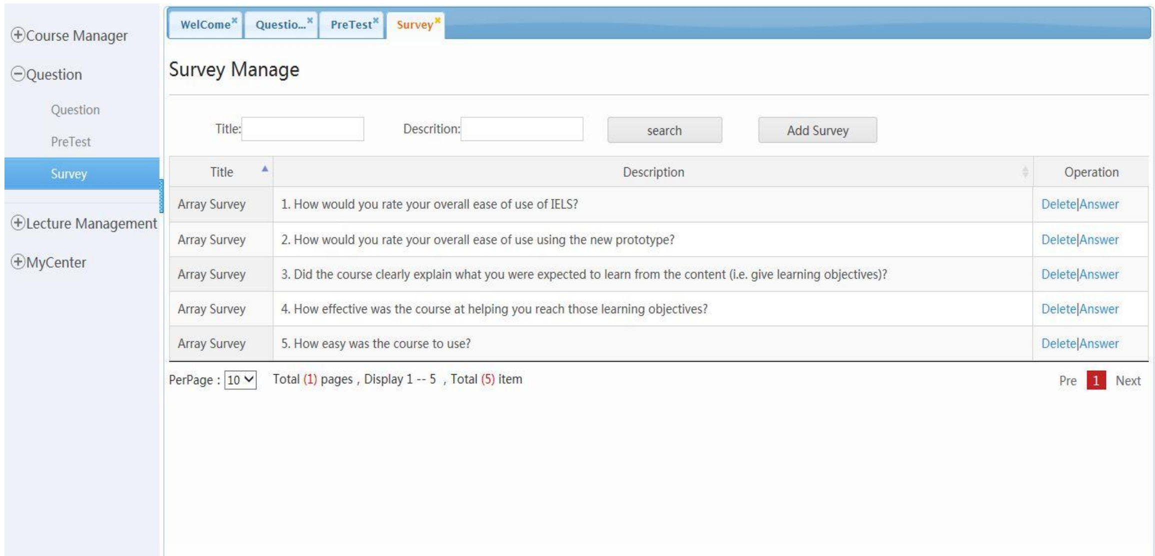


Fig. 17. Survey Management Screen

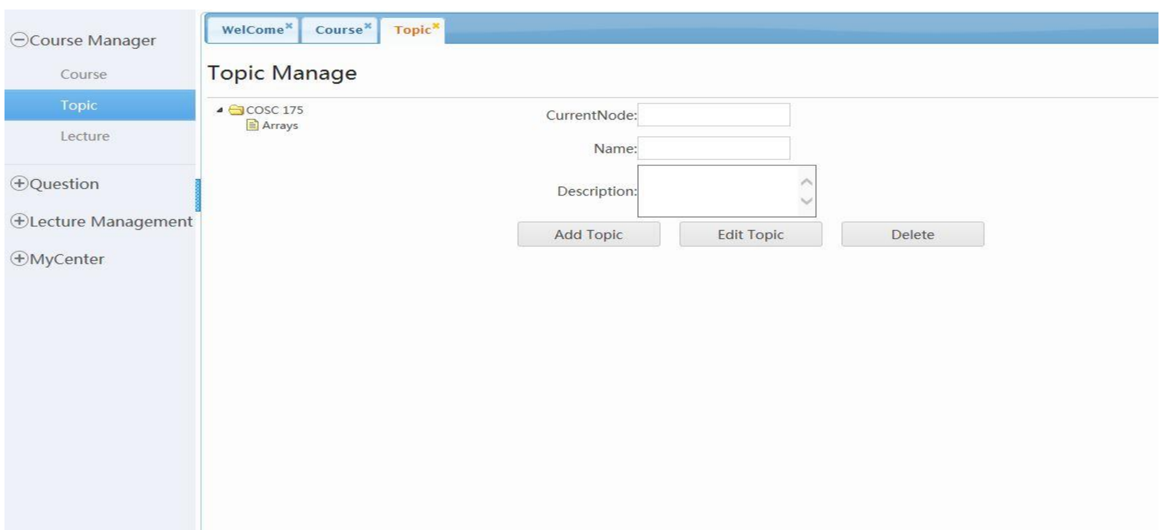


Fig. 18. Lecture Structure Management Screen

5.3 Student GUI

The student GUI allows the student to view lectures and receive feedback, providing an interactive way to communicate with the system. IELS has a vector placed between the student's GUI and other components of the IELS. It collects, records, and measures the student's learning activity and performance during a lecture session as well as the student's statistic data. Figures 19—24 is presenting screen shots of the student login.

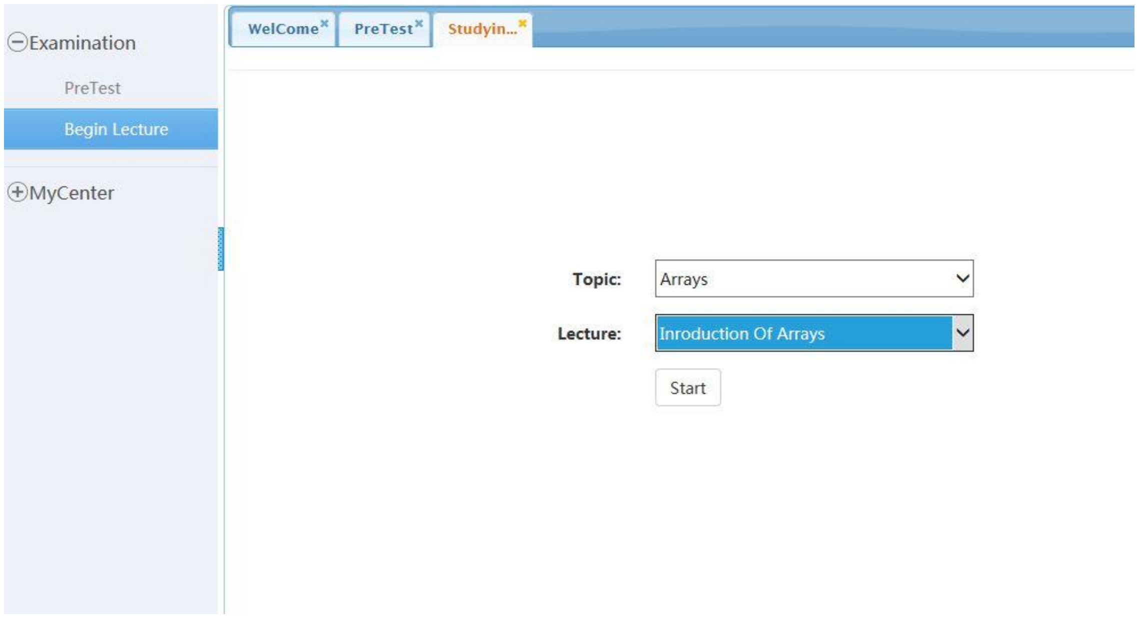


Fig. 19. Student Lecture Study Screen

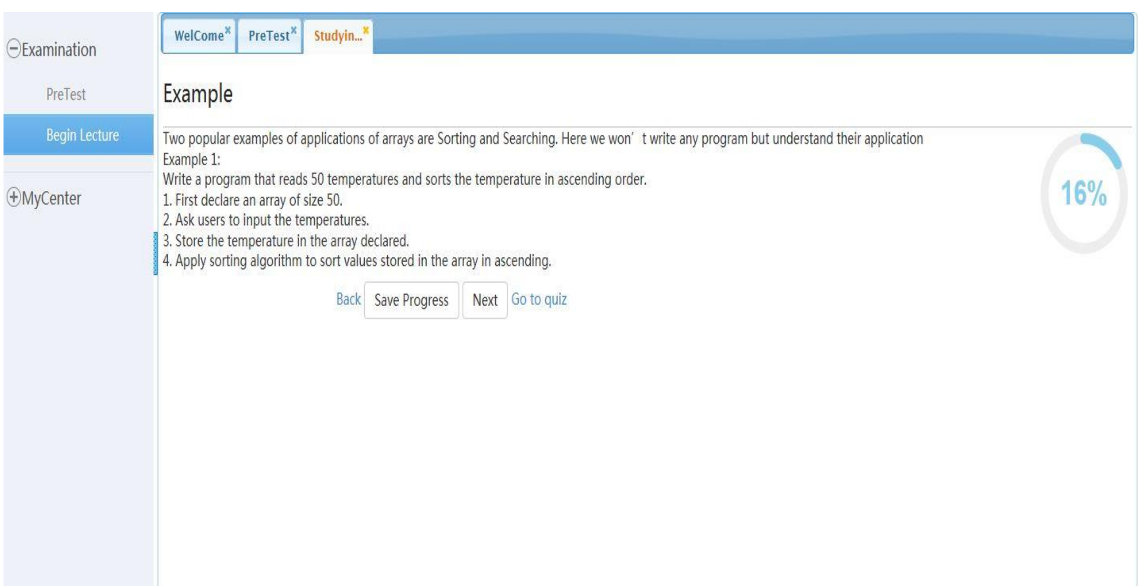


Fig. 20. Student Read Exmaple Screen

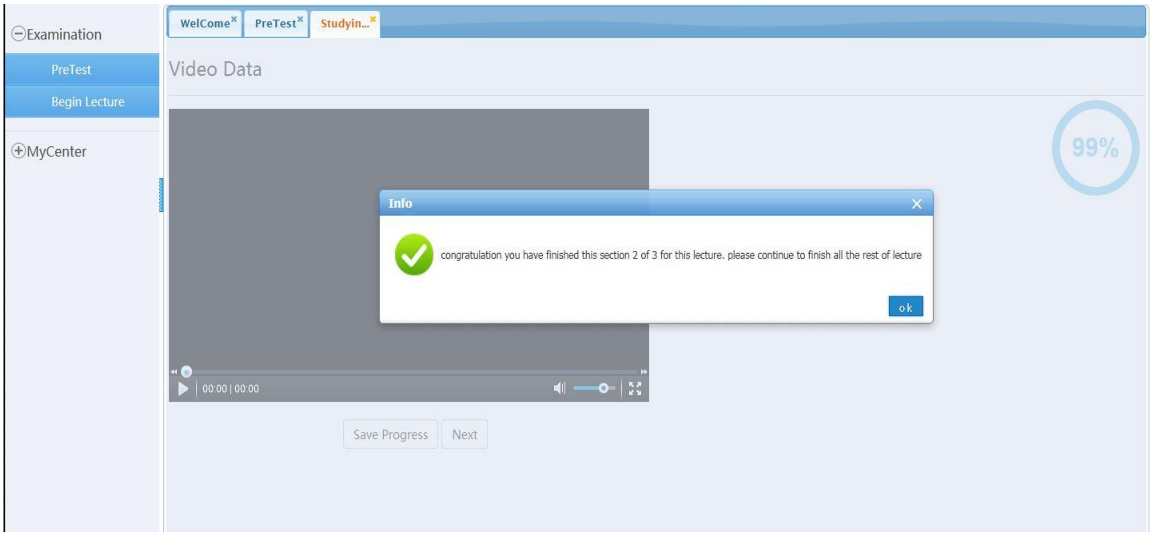


Fig. 21. Notification of Finish Lecture

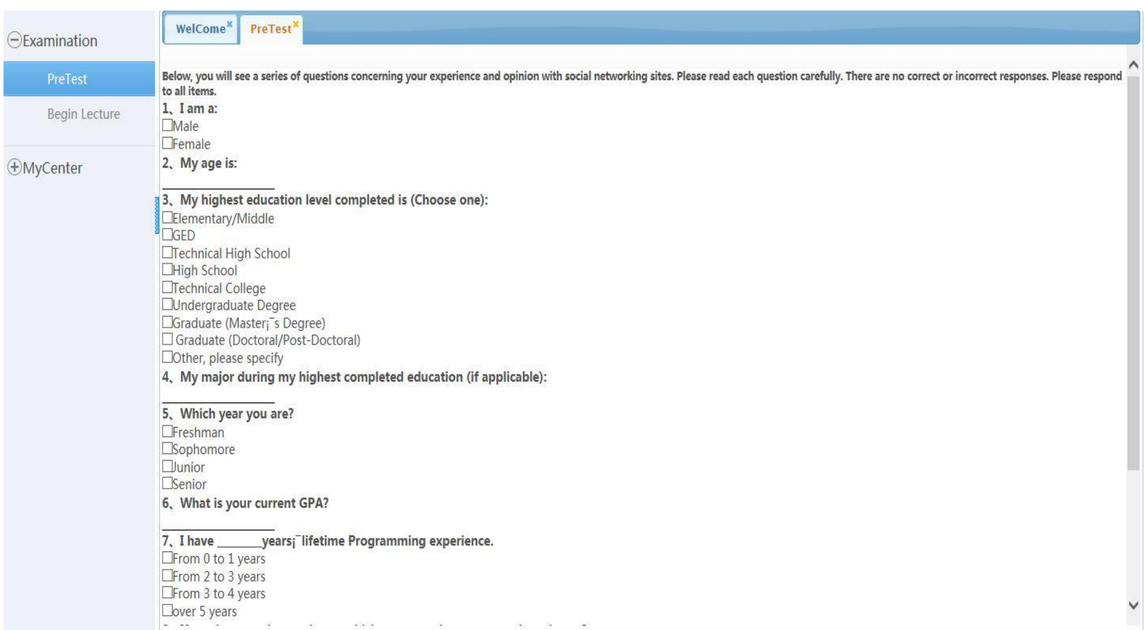


Fig. 22. Pre_Test Screen

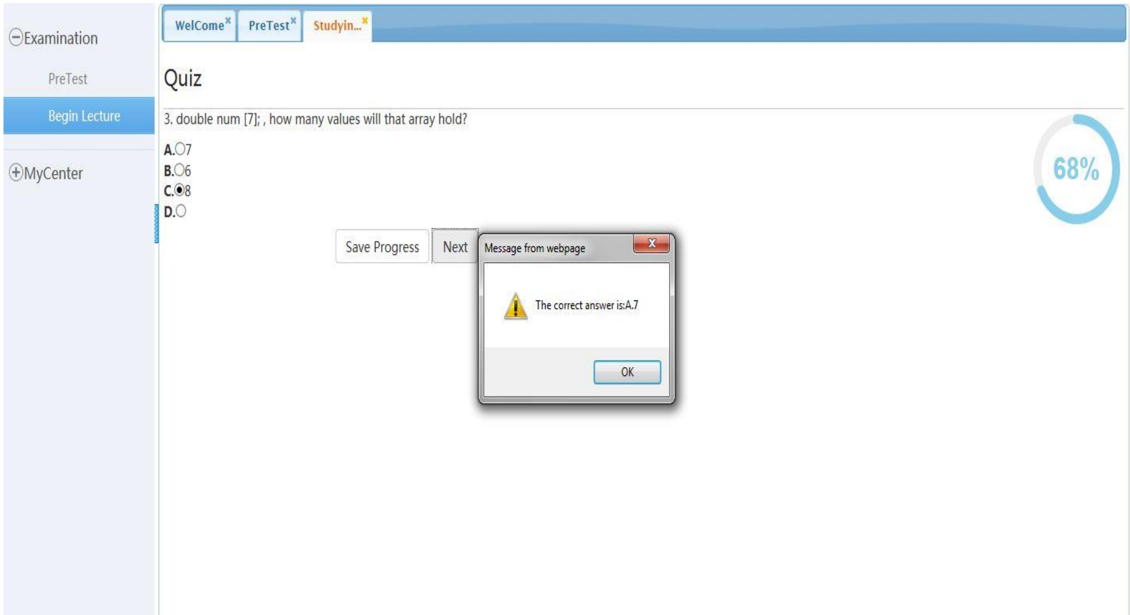


Fig. 23. Notification of Incorrect Answer

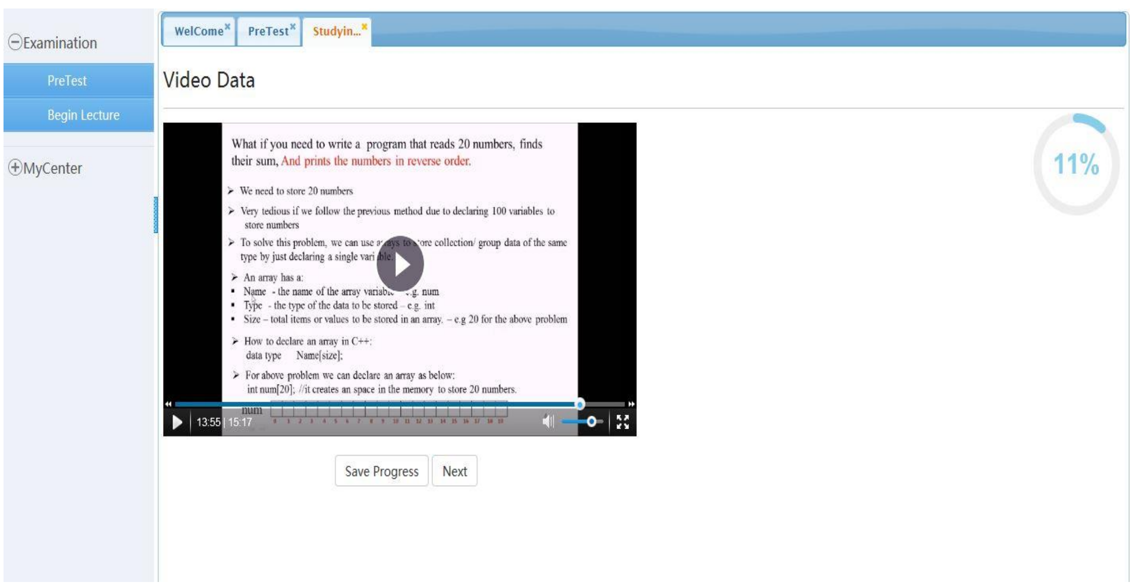


Fig. 24. Lecture in Video Format

6. Research Questions

6.1 Research Questions

This dissertation focuses on the following research questions:

- How to adaptively select teaching materials and to provide different personalized lectures for students with diverse backgrounds and intellectual capabilities?
- How to personalize curriculums and provide tailored lessons in order to increase students' performance on standardized tests?

In what follows, we discuss these two issues in details.

6.1.1 Personalized Lecture Delivery

An effective teaching system must be able to customize lesson planning and provide a personalized pathway to the needs of the students. Traditional classroom teaching and most e-learning systems do not pay attention to individual needs and are unable to offer lessons geared toward students' different background and abilities. These lectures are done in prefixed time frame regardless different learning curves of the students. Good students may only need to have a portion of the time to understand the materials while students with below the average level learning capability may spend much longer to grasp the topic. The "one-size-fit-all" lectures need to be improved to achieve better effectiveness and efficiency for every student. An effective teaching approach promises to spend the shortest time to cover most topics for good students while enough time for others to understand the same topics. The eventual learning objectives must be met but the path and time to reach them may vary from student to student. The "No Student Left Behind" strategy requires the e-learning system constantly pays attention to the student's progress and feedbacks, controls the difficult level of lessons, and selects appropriate examples and exercises. The adjustment of the teaching materials and sequences must be done in real time and in response to the student's performance during a session, not before and after the lecture.

6.1.2 Personalized Curriculum Construction

There are many test prep training programs and software on the market targeting the tests such as entrance examinations to colleges (SAT and ACT), Business schools (GMAT), Law schools (LSAT), medical schools (MCAT) and graduate schools (GRE). The students who take test prep courses do not want an e-learning system to cover every topic from A to Z in a subject area. Instead, a personalized curriculum and the lectures designed based on it are considered desirable. Unlike e-learning systems that provide college courses for students, the curriculums of test preparations do not have to be the same for every student. There is no need to cover every subject in details. On the contrary, a personalized curriculum enables the student to focus her/his weaknesses while skipping the topics that have been understood. An intelligent e-learning system designs a set of diagnostic questions using the guidance of a standardized test curriculum and helps identify a student's strengths and weaknesses. For well understood concepts and subjects, the learning system should skip them and design lecture materials for the topics that the students are having problems with. Depending on the scores of diagnostic tests, some subjects may be discussed briefly as a way of review if the student made some mistakes on the test but did demonstrate some knowledge, while other subjects on which the student did not do well at all, an e-learning system should provide detailed explanations and set of comprehensive examples and exercises to fill their knowledge gap. This is an effective way for students to realize the great point growth and to nail down his/her target scores with the shortest time possible.

In order to answer these questions and provide solutions to individualized lesson delivery, this research proposes, designs, develops and tests IELS capable of conducting intelligent and effective teaching to students with diverse backgrounds and learning abilities. IELS is also able to construct individualized curriculums and to tailor learning pathways to the needs of students in order to hit their target scores on standardized tests.

6.2 Experiment

Based on research questions there are two goals of the experiments:

1) To study the time needed to understand a set of learning objectives in order to demonstrate students with different backgrounds may take different lengths of time to complete a learning unit.

2) To study the advantages of IELS compared with a traditional online learning system in term of learning curve and efficiency.

6.2.1 Methodology

A quasi-experiment was conducted in one section of a COSC175 course (programming logic using C++) in the summer of 2016 at Towson University, using a pretest-posttest control group design. Both course sections were taught by the instructor. The students were tested on the programming construct Arrays in C++.

One of the sections used the IELS Dynamic version (treatment group) and the other used the traditional E-learning version (control group). The study was conducted during laboratory sessions, which occurred at different times for each section. Both groups were given a pre-test at the beginning and a post-survey at the end of the session. A total of 33 students participated in the study. Both sections' student demographic data are presented in Table 2.

	treatment		control	
	n	%	n	%
Gender				
Male	12	0.6	10	0.56
Female	8	0.4	8	0.44
Student Standing				
Freshman	7	0.35	4	0.22
Sophomore	7	0.35	3	0.17
Junior	4	0.2	4	0.22
Senior	2	0.1	7	0.39
Major				
Computer Science	18	0.9	15	0.83
Non-Computer Science	2	0.1	3	0.17

Table 2 Student Demographics

Both the pre-survey and post-survey included multiple-choice questions related to student demographics and diagnostic questions, including self-assessment and experiences. The lecture was developed by a computer science education researcher from Towson University. The surveys were embedded in the web-based learning system and the data was collected and stored in the database. The learning system was deployed on www.adaptiveelearn.com.

The learning content is divided into three segments, and each segment includes advanced, intermediate and introductory sections. As mentioned in the previous section, the module (segment) is internally exchangeable.

For example, if student 1 started with an advanced segment but struggled to follow that segment, then he would be given the course content at the intermediate level. Similarly, if student 2 started with an introductory content but his real-time performance indicated he mastered the topic very quickly, then he would be given the teaching materials at the advanced level.

The most important reason for having the module component in the system is to allow flexibility and feasibility to dynamically deliver course content in an adaptive and continuous manner. The three sections developed in Arrays include:

1) Introduction of array

2) Declaration

3) Two-dimension arrays

The tutorial material in repository is in video format, the examples are in text format, and the quizzes are in multiple-choice question format.

Based on the pre-survey and the post-survey scores, the following set of hypotheses were proposed to compare IELS Dynamic version (treatment) and IELS Static version (control) on the following dependent variables: amount of time spent reading examples, amount of time spent reading material, and difficulty level.

A set of hypotheses to compare IELS Dynamic version (treatment) and the IELS Static version (control) were proposed on the following dependent variables: average score and average time length of tutorial section.

H1:	Students will take different amounts of time to read examples in the treatment groups.
Rationale:	As discussed previously, an advanced student could skip some examples to speed up his session, while a student at the introductory level may need to see more examples and spend more time on lecture materials in order to understand a new concept.
H2:	The IELS can constantly pay attention to the student's progress and feedback and can control the difficultly level of lessons and select appropriate teaching material.
Rationale:	The use of a knowledge base enables the system to align materials accurately with the realities of the students' different levels of comprehension, progress and weaknesses. Since each module has a different difficulty level, a different tutorial video will be triggered and the length of each video will also be different.
H3:	By using IELS, the student will have variant difficulty levels of material in different learning segments.
Rationale:	As previously discussed, the knowledge base analyzes the student's intellectual ability and background and makes a judgment on the difficulty level of course content.
H4:	There is a significant difference in the task completion time between individuals who use the IELS and those who use the regular E-learning system.
Rationale:	IELS evaluates the students' real-time learning activity, determines their competency level, analyzes their progress, and selects appropriate teaching materials. This system should also reduce the time it takes different students to complete a lecture.

Table 3 Hypotheses and Rationale

6.2.2 Experiment Results

In summer 2016, IELS was first deployed at www.adaptiveelearn.com. The Array section of COSC175 was the first course for testing IELS. Further, the programming background of these students was highly diverse. Some incoming freshmen have had some programming in their high schools; others have had none. Thus, it is important for the course to be able to adapt to the different student aptitude levels and motivations. Therefore, this C++ programming course is thus well-suited for evaluating the IELS application.

Hypotheses 1, 2, and 3 address Research Question 1 and, Hypothesis 4 addresses Research Question 2.

Fig.24 shows the number of seconds each student in the treatment group spent in the example section of lecture. Students interacting with the IELS spent different amounts of time because the IELS provided different appropriate examples for each student based on their real-time performance. This leads us to accept H1

Student Review Example Time

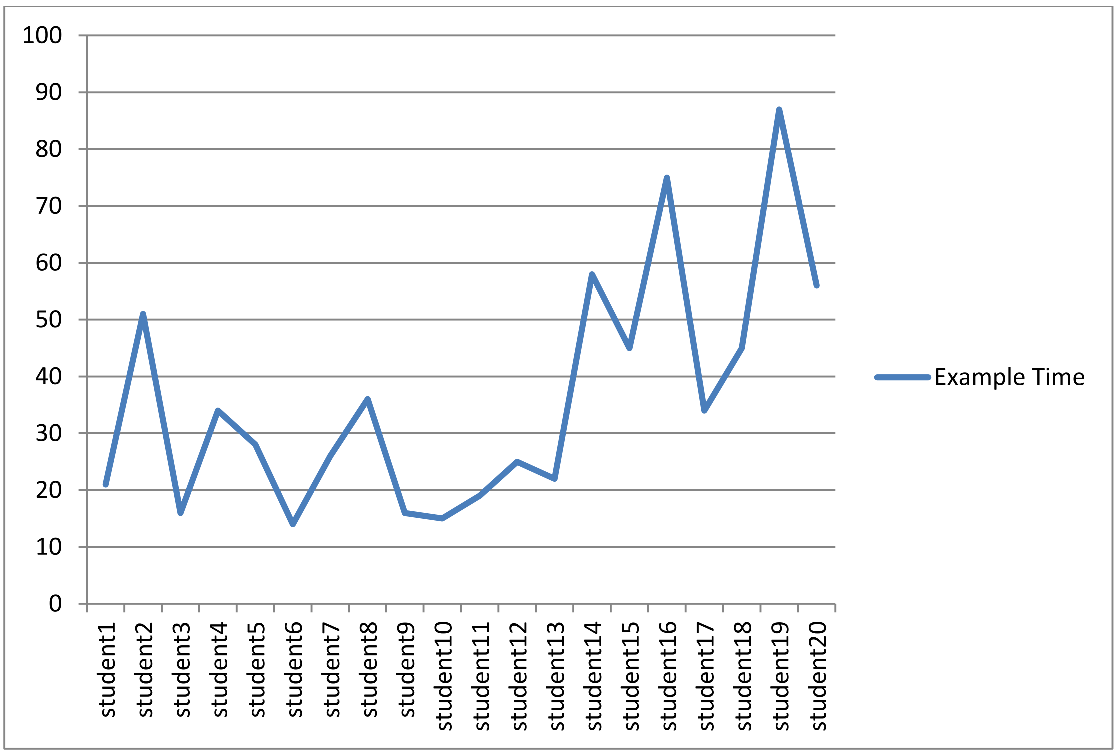


Fig. 25. Time students spent reviewing examples

Figure 25 shows the number of seconds students in the control group spent reviewing the example section. Figure 26 shows each student's post-test score. Students interacting with the IELS spent different amounts of time reviewing teaching materials, and they all met the learning objectives. H2 has indicated that learning objectives must be met, but the path and time to reach them may vary from student to student.

The No Student Left Behind strategy requires the E-Learning system to constantly pay attention to the student's progress and feedback, control the difficulty level of lessons, and select appropriate teaching material. The teaching material is designed in video format, which allows students to rewind or escape parts of the video if they understand the concept.

Based on their comprehension and background, the time spent on teaching material will vary for different students. Since there are three different levels on each module, the content will vary for each student on each segment based on their comprehension and competence. This leads us to accept H2 (see Figure 26 and Figure 27).

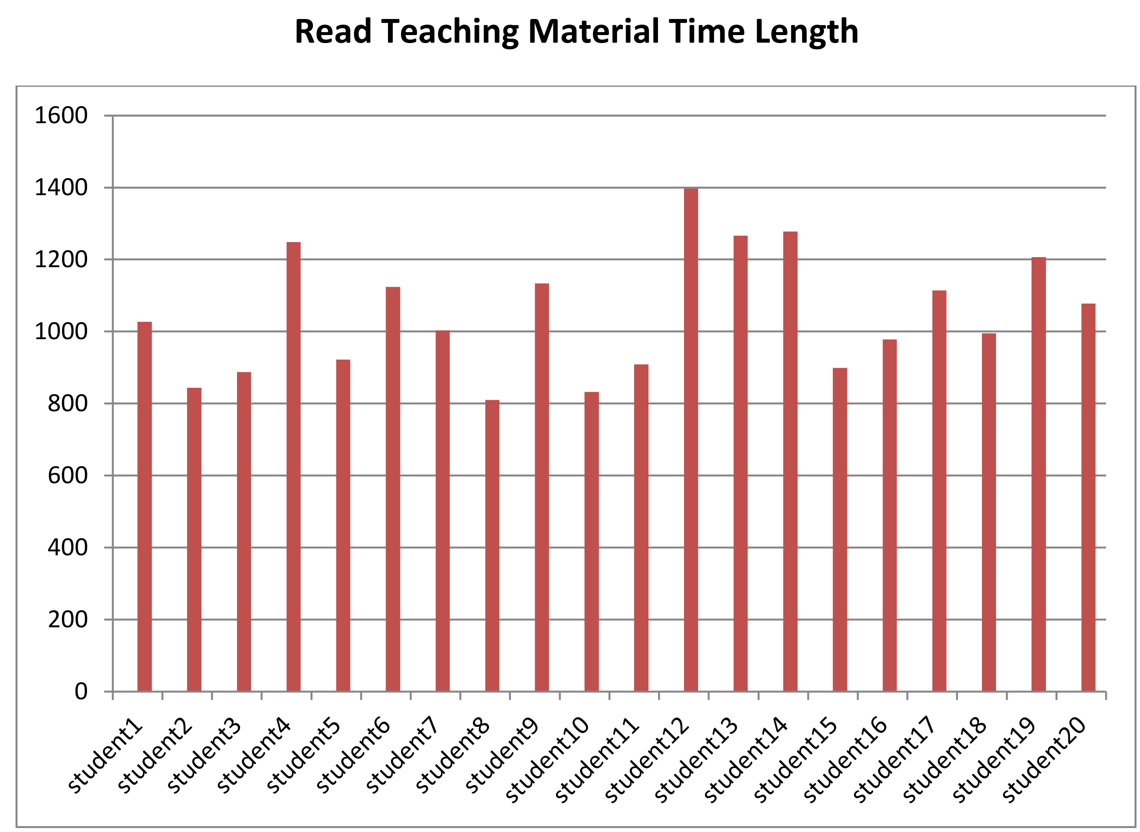


Fig. 26. Time students spent reviewing teaching Material

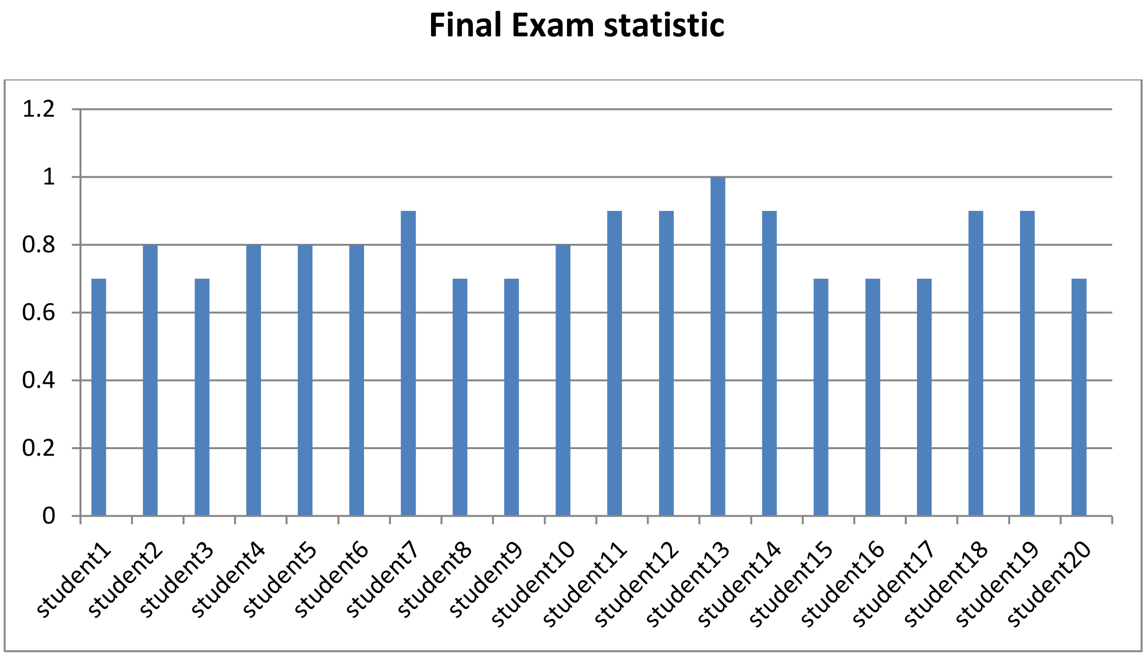
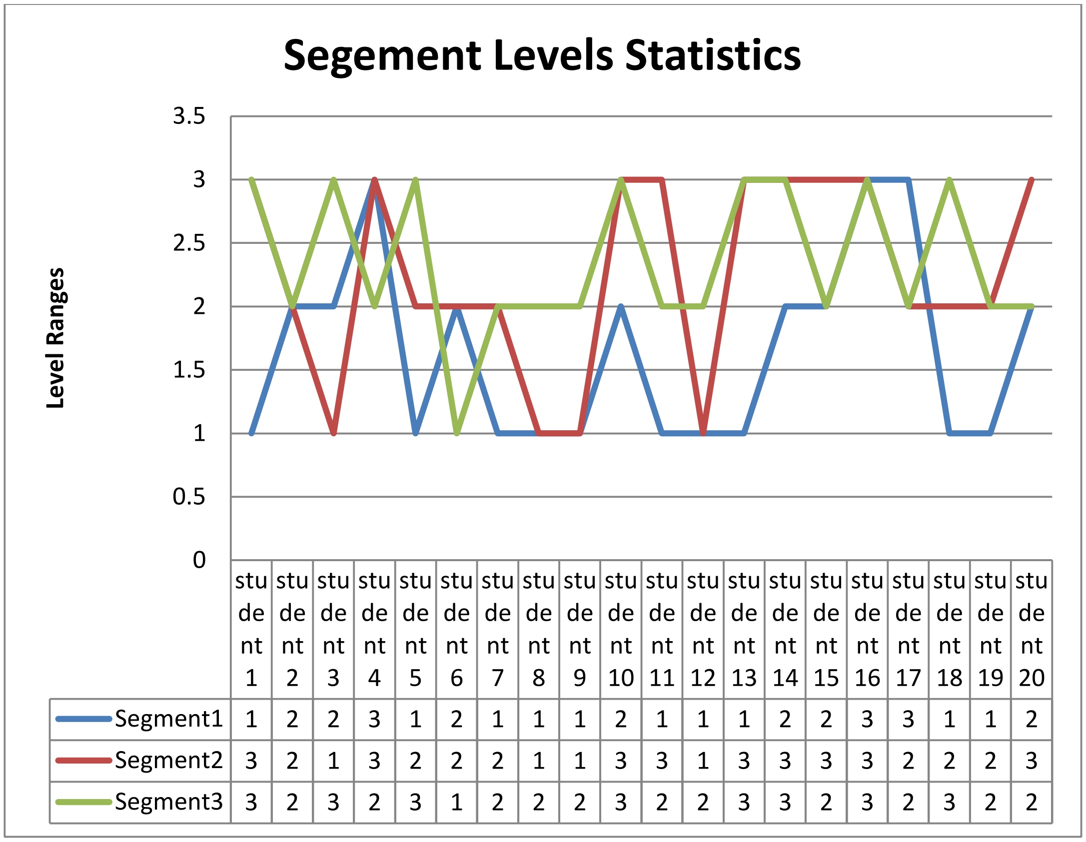


Fig. 27. Final Score for Each Student in Lecture of Introduction of Array

In other words, the end of the module n -1 lecture is always followed by the first tutorial segment in module n. Let us assume that a student is working on module 1, any module 2 with a difficulty level of advanced, intermediate, or introductory can be selected to provide a continuous lecture. Figure 27 can list each student in different difficulty modules. We can accept H3.



	student 1	student 2	student 3	student 4	student 5	student 6	student 7	student 8	student 9	student 10	student 11	student 12	student 13	student 14	student 15	student 16	student 17	student 18	student 19	student 20
Segment1	1	2	2	3	1	2	1	1	1	2	1	1	1	2	2	3	3	1	1	2
Segment2	3	2	1	3	2	2	2	1	1	3	3	1	3	3	3	3	2	2	2	3
Segment3	3	2	3	2	3	1	2	2	2	3	2	2	3	3	2	3	2	3	2	2

Fig. 28. Students in various modules with different difficulty levels

The treatment group was using the IELS with knowledge base reasoning algorithm, and the control group was using a traditional teaching approach, which set the difficulty level on intermediate. Figure 29 shows that the average score of each teaching session is similar, but the amount of time in the teaching section shows a big difference. The No Student Left Behind teaching strategy was implemented on both groups.

Traditional E-learning approach VS IELS

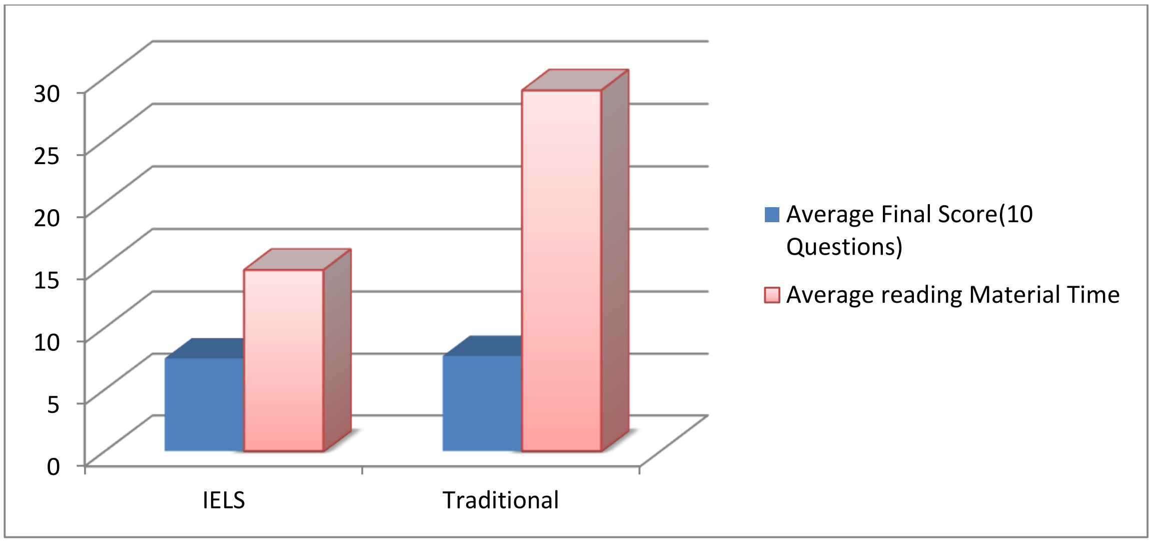


Fig. 29. Comparision on Traditional Teaching Environment and IELS

Since the most widely adopted statistical procedure for comparing two means is the T-test, I used this procedure to compare differences in the task-completion time between individuals who use the IELS and those who use regular E-learning system. The T-test from SPSS will be applied on this experiment to test this hypothesis by recruiting two groups of participants. One group used the IELS with knowledge base reasoning algorithm, and the other group used a traditional teaching approach with same teaching material. The two groups are reasonably independent from each other.

The T-test procedure was selected for those two groups. Zero represented the participants who completed the tasks without IELS, and one represented the participants who completed the tasks with IELS. Below are the results from the T-test. The Fig. 29 completion time is in seconds.

T-Test

[DataSet1] C:\Users\dehui-li\Desktop\Finalzhou\experiment\spss20161010.sav

Group Statistics

	group	N	Mean	Std. Deviation	Std. Error Mean
CompletionTime	Traditional E-learning	18	1678.7222	134.55670	31.71532
	IELS_user	20	1092.6000	135.13362	30.21680

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
CompletionTime	Equal variances assumed	.127	.724	13.377	36	.000	586.12222	43.81557	497.26012	674.98433
	Equal variances not assumed			13.380	35.614	.000	586.12222	43.80544	497.24725	674.99719

Fig. 30. T-Test Result for IELS verse Traditional Teaching Approach

Since $p < .0001$ is less than our chosen significance level $\alpha = 0.05$, we can reject the null hypothesis and conclude that the mean completion time of IELS users and non-IELS users is significantly different.

By running the SPSS T-test we can see an independent-samples T-test suggests that there is a significant difference in the task-completion time between the group that used the IELS and the one that used regular E-learning system the prediction software ($t(36)= 13.377$, $p < 0.05$).

Based on both Figure 29 and the T-test results, IELS makes it possible for students to learn new concepts at their own speed and level of comprehension by providing lecture content at different levels. Some students can finish the lecture quickly, while others at the introductory level may take longer. With detailed explanations and more examples, these students, hopefully, would eventually achieve the same lecture objectives. This leads us to accept H4.

Overall, IELS is responsible for constructing a new lecture unit. It receives recommendation from the knowledge base and the case base. The knowledge base analyzes the student's intellectual ability and background and makes a judgment on the difficulty level of course content. The main factors to be considered include the student's self assessment, maturity, familiarity to the subject being discussed, and his past academic performance. The case base suggests a set of examples and exercises that have been proven to be effective for students with similar characteristics.

Put simply, IELS assumes the role of a hypothetical teacher, who presents teaching materials to an individual student. Based on the observation and the student's feedback, it is able to dynamically choose lecture contents and to intelligently select appropriate examples and exercises to reduce the learning curve and achieve greater student comprehension.

6.3 Contribution of IELS

Through the evolution and acceptance of computer-based training and distributed computing systems, the technology that has lagged behind is the ability of instructional delivery systems to systematically adapt to a learner's acquisition of knowledge. This adaptive responsiveness can consistently deliver the benefits of an instructor's individualized attention across the entire learner population.

To achieve this adaptation on behalf of the learner across different topics effectively, requires both the breadth of material and reactive responsiveness of an intelligent learning system. An intelligent learning system that can continually learn from the students interacting with the content to alter the information presented, tailor the exercises to assist knowledge acquisition, and customize quizzes that feedback performance to the student, instructors and subject matter experts, and enhances the adaptation of the learning system itself.

The benefit of this system is achieving the desired learning outcomes across a diverse student population while reducing the time investment for intermediate and advanced learners and students who have a talent for rapid knowledge acquisition. Aside from college courses, where this type of intelligent, effective, and distributed digital learning system can address the varying knowledge levels of incoming students, there is a realm of standardized testing, both academic and professional, that would benefit from presenting a breadth and diversity of content designed to elevate individual performance.

The benefits of minimizing time investment as related to performance are realized by both student, in an academic setting and testing subject, in a professional environment.

An advanced student can minimize their interaction with the course materials to review and efficiently prepare for the test. Intermediate learners can benefit through review of additional materials to elevate their performance to an advanced level before testing. The magnitude of potential that can be realized from a beginning learner that might achieve intermediate or even advanced status is where an intelligent and effective E-learning system (IELS) rewards the considerable technology and academic investment.

This report delineates the successful experiment of an IELS utilized in teaching a section of a beginning C++ programming course at Towson University. However similar results and benefits would be realized in using IELS for:

Professional test preparation for licensure, such as CPA, bar or board examinations

Student preparation for standardized testing, as administered to measure performance to meet Every Student Succeeds Act (ESSA) assessments

Preparation for entry testing for graduate studies, such as medical schools (MCAT), law schools (LSAT), or business schools (GMAT)

In each of these situations the learner could benefit from an adaptive and responsive learning system that allows maximum knowledge acquisition with a time investment appropriate to their needs.

7. Conclusions and Suggestions for Future Research

7.1 Conclusion

The purpose of this research was to develop, implement, and evaluate IELS in order to provide tailored lessons to every student based on their levels of comprehension, progress, and weaknesses. By analyzing the student’s statistical data in his background and intellectual ability and utilizing dynamic data collected during a lecture session in real time, IELS is able to provide personalized lessons with different levels of difficulty for students with diverse backgrounds. IELS evaluates the student’s real-time learning activity, determines their competency level, analyzes their progress, and selects appropriate teaching materials. IELS not only applies to E-learning, but it can also can be applied to entrance examinations to colleges (SAT and ACT), business schools (GMAT), law schools (LSAT), medical schools (MCAT) and graduate schools (GRE).

IELS was designed and implemented to answer the following research questions:

How to adaptively select teaching materials and to provide different personalized lectures for students with diverse backgrounds and intellectual capabilities?

How to personalize curriculums and provide tailored lessons in order to increase students' performance on standardized tests?

7.2 Suggestions for Future Research

After a significant growth in cloud-based systems, many industries gave their attention to cloud-computing solutions. E-learning is a promising application area since its typical requirements are around a dynamic allocation of resources. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service-provider interaction.

For further research, researchers can try to implement a public cloud, where learners can access such laboratories to support their practical learning without the need to set up laboratories at their own institutions. The cloud infrastructure enables multiple laboratories to come together virtually to create an ecosystem for educators and learners. From such a system, educators can pick and mix materials to create suitable courses for their students, and the learners can experience different types of devices and laboratories through the cloud. The private information and data-source security in the cloud-computing environment will need to be another research topic in the future.

Appendices

Appendix A – Institutional Review Board Documents

IRB Protocol Approval # 1607003172

TL Taylor, Amy L.
Today, 1:22 PM
Li, Dehui*; Zhou, Harry

Reply all

Flag for follow up. Start by Wednesday, November 09, 2016. Due by Wednesday, November 09, 2016.

Dear
The IRB has approved your protocol "Intelligent Online Course Delivery System " effective 7/31/2016 as Exempt Research, Category 2.

Your IRB protocol can now be viewed by your faculty advisor in MyOSPR. For more information, please visit: http://www.towson.edu/academics/research/sponsored/myospr.html

If you should encounter any new risks, reactions, or injuries to subjects while conducting your research, please notify IRB@towson.edu. Should your research extend beyond one year in duration, or should there be substantive changes in your research protocol, you will need to submit another application.

We are offering training and orientation sessions for faculty in the fall, I encourage you to sign up for one of the sessions:
http://fusion.towson.edu/www/signupGeneric/index.cfm?type=OSPR

Regards,
Towson IRB

Amy L. Taylor, MBA, CRA · Assistant Vice President for Research
Office of Sponsored Programs & Research
Towson University · 8000 York Road · Towson, Maryland, 21252-0001
t. 410-704-4931 · f. 410-704-4494

TOWSON UNIVERSITY

Appendix B – Consent Form

This is a research project being conducted by Dehui Li in the Computer and Information Sciences Department at Towson University. The title of this study is IELS (intelligent online course delivery system).

The purpose of this research project is to assess the current social media site's mobile device interface for usability and accessibility for elder adult users. We hope that in the future, older adults might benefit from this study through an improved interface for these types of sites.

The purpose of this experiment is to evaluate usability of IELS (intelligent online course delivery system). This is part of an Adaptive E-learning research program aimed at analyzing the efficiency, effectives and student satisfaction of IELS.

As a participant, you will be invited to participant in a survey and tasks regarding your experience with IELS (intelligent online course delivery system)

All information will remain strictly confidential. Although the descriptions and findings may be published, at no time will your name be used. There are no known risks with participating in this research project.

Your participation in this research is completely voluntary. You may choose not to take part at all. If you decide to participate in this research, you may stop participating at any time. If you decide not to participate in this study or if you stop participating at any time, you will not be penalized or lose any benefits to which you otherwise qualify.

This research is being conducted by Dehui Li and Dr. Harry Zhou in the Department of Computer and Information Sciences at Towson University. If you have any questions about the research study itself, please contact Dehui Li at 443-2482100, or Towson University's Institutional Review Board for the Protection of Human Participants, irb@towson.edu at (410) 704-2236. Thank you for your time and willingness to participate in this experiment.

Your signature indicates that:

• The research has been explained to you;

• Your questions have been fully answered; and

• You freely and voluntarily choose to participate in this research project

Signature: ___________________________ Date: ______________________

Witness: ____________________________Date: ______________________

THIS PROJECT HAS BEEN REVIEWED BY THE INSTITUTIONAL REVIEW BOARD FOR THE PROTECTION OF HUMAN PARTICIPANTS AT TOWSON UNIVERSITY.

Appendix C – Assessment for Student Learning

APPENDICES

APPENDIX 1: Pre-Survey

Below, you will see a series of questions concerning your experience and opinion with social networking sites. Please read each question carefully. There are no correct or incorrect responses. Please *respond to all items*.

1. I am a:
____ Male
____ Female

2. My age is: ________

3. My highest education level completed is (Choose one):
____ Elementary/Middle
____ GED
____ Technical High School
____ High School
____ Technical College
____ Undergraduate Degree
____ Graduate (Master's Degree)
____ Graduate (Doctoral/Post-Doctoral)
____ Other, please specify

4. My major during my highest completed education (if applicable):
____

5. Which year you are?
____ Freshman
____Sophomore
____ Junior
____ Senior

6. What is your current GPA? _________________________

7. I have _________years' lifetime Programming experience.
   ____ From 0 to 1 years
   ____ From 2 to 3 years
   ____ From 3 to 4 years
   ____ over 5 years

8. If you have work experience, which computer language you have learnt?

_____ C#
_____ Visual Basic
_____ Java
_____ C++
_____ Python
_____ Others

APPENDIX 2: Post survey

1. How would you rate your overall ease of use using the ITS?
_____ Very difficult
_____ Difficult
_____ Neutral
_____ Easy
_____ Very Easy

2. How would you rate your overall ease of use using the new prototype?
   a. _____ Very difficult
   b. _____ Difficult
   c. _____ Neutral
   d. _____ Easy
   e. _____ Very Easy

3. Did the course clearly explain what you were expected to learn from the course (i.e. give learning objectives)?

   a. _____ yes

   b. _____NO

   c.

4. How effective was the course at helping you reach those learning objectives?

   a. _____ Not at all
   b. _____ Not very
   c. _____ Mediocre
   d. _____ quite
   e. _____ Very

5. How easy was the course to use?
   a. _____ Not at all

b. _____ Not very
c. _____ Mediocre
d. _____ quite
e. _____ Very

6. How engaging you found the course?

   a. _____ Not at all
   b. _____ Not very
   c. _____ Mediocre
   d. _____ quite
   e. _____ Very

7. How visually attractive (pleasing? Seductive?) You found the course?

   a. _____ Not at all
   b. _____ Not very
   c. _____ Mediocre
   d. _____ quite
   e. _____ Very

8. Was the information in the course easily understandable?
   a. _____ Not at all
   b. _____ Not very
   c. _____ Mediocre
   d. _____ quite
   e. _____ Very

9. Was further guidance offered where information was complex?
   a. _____ Not at all
   b. _____ Not very
   c. _____ Mediocre
   d. _____ quite
   e. _____ Very

List of References

Akbulut. Y., & Cardak, C. S. (2012). Adaptive educational hypermedia accommodating learning styles: A content analysis of publications from 2000 to 2011. Computers & Education, 58(2), 835-842.

Anderson, J. R., Boyle, C. F., Corbett, A., and Lewis, M. W. (1990) Cognitive modelling and intelligent tutoring. Artificial Intelligence, 42, 7-49.

Anderson, J. R., Corbett, A. T., & O'Brien, A. T. (1995). Student modeling in the ACT programming tutor. In P Nichols, S. Chipman, & R. Brennan (Eds.), Cognitively diagnostic assessment (pp. 19-41). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Aroyo, L., Dolog, P., Houben, G-J., Kravcik, M., Naeve, A., Nilsson, M. & Wild, F. (2006). Interoperability in personalized adaptive learning. Educational Technology & Society, 9(2), 4-18.

Azough, Bellafkih, Bouyakhf, 2010. Adaptive E-learning using Genetic Algorithms. International Journal of Computer Science and Network Security, VOL.10 No.7. National Institute of posts and Telecommunications Rabat, Morocco - Azough, Bellafkih; Faculty of Sciences Rabat Agdal, Morocco - Azough, Bouyakh

Baffes, P. T., & Mooney, R. J., (1996). A novel application of theory refinement to student modeling. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 403, 408.

Bajraktarevic, N., Hall, W., & Fullick, P. (2003). Incorporating learning styles in hypermedia environment: Empirical evaluation. In P. de Bra, H. C. Davis, J. Kay & M. Schraefel (Eds.), Proceedings of the Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, Nottingham, UK: Eindhoven University, 41-52.

Bangert-Drowns, R. L., Kulik, J. A., & Kulik, C. - L. C. (1985). Effectiveness of computer-based education in secondary schools. Journal of Computer-Based lnstruction, 12, 59-68.

Baxter, E. 1990. Comparing Conventional and Resource Based Education in Chemical Engineering: Student Perceptions of a Teaching Innovation, Higher Education, 19:323-340.

Bloom, B.S. Mesia B.B. & Krathwohl, D. R. (1964). Taxonomy of Educational Objectives (two vols: The Affective Domain & The Cognitive Domain). New York: david McKay.

Bloom, B. S. (1984) The 2-sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13, pp. 4-16.

Brown, E., Fisher, T., & Brailsford, T., (2007). Real users, real results: examining the limitations of learning styles within AEH. In: Proceedings of the Eighteenth ACM Conference on Hypertext and Hypermedia , 2007, . pp. 57-66. http://doi.acm.org/10.1145/1286240.1286261

Brusilovsky, P. and Miller, P. (2001) Course Delivery Systems for the Virtual University. In: T. Tschang and T. Della Senta (eds.): Access to Knowledge: New Information Technologies and the Emergence of the Virtual University. Amsterdam: Elsevier Science, pp. 167-206.

Carver, C. A., Jr., Howard, R. A., & Lane, W. D. (1999). Enhancing student learning through hypermedia courseware and incorporation of student learning styles. IEEE Transactions on Education, 42(1), 33–38.

Cassell, J., Sullivan, J., Prevost, S., & Churchill, E. (2000). Embodied Conversational Agents. Cambridge: MIT Press.

Chen, S. Y. and Liu, X. (2008) An Integrated Approach for Modeling Learning Patterns of Students in Web-based instruction: A Cognitive Style Perspective. ACM Transactions on Computer-Human Interaction. 15(1), Article 1.

Chen, S. Y., & Macredie, R. D. (2002). Cognitive styles and hypermedia navigation: Development of a learning model. Journal of the American Society for Information Science and Technology, 53(1), 3-15.

Chen, S. Y., Fan, J.-P., & Macredie, R. D. (2004). Navigation in hypermedia learning systems: Experts vs. novices. Computers in Human Behavior, 22(2), 251-266.

Chiou, C. K., Tseng, Judy C. R., Hwang, G. J., & Heller, S. (2010). An adaptive navigation support system for conducting context-aware ubiquitous learning in museums. Computers & Education, 55(2), 834-845.

Coffield, F., Moseley, D., Hall, E., & Ecclestone, K. (2004). Learning styles and pedagogy in post-16 learning: A systematic and critical review. London, UK: Learning and Skills Research Centre.

Conati, C., A. Gertner, et al. (2002). “Using Bayesian Networks to Manage Uncertainty in Student Modeling.” Journal of User Modeling and User-Adapted Interaction 12(4): 371-417.

Corbett, A. T. and Anderson, J. A. (1992) Knowledge tracing in the ACT programming tutor. In: Proceedings of 14-th Annual Conference of the Cognitive Science Society.

DeHui. L., Zhou, H. (2015). Individualized Tutoring System Provides Tailored Learning Experiences, Towson University.

Elorriaga, J. A., & Fernández-Castro, I. (2000). Using case-based reasoning in instructional planning. Towards a hybrid self-improving instructional planner. International Journal of Artificial Intelligence in Education, 11, 416-449.

Evans, C., & Waring, M. (2011). Student teacher assessment feedback preferences: The influence of cognitive styles and gender. Learning and Individual Differences Journal, 21(3), 271-280.

Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. Engineering Education, 78(7), 674-681.

Felder, R. M., & Soloman, B. A. (1997). Index of learning style questionnaire. Retrieved from http://www.engr.ncsu.edu/learningstyles/ilsweb.html

Filippidis, S. K., & Tsoukalas, I. A. (2009). On the use of adaptive instructional images based on the sequential-global dimension of the Felder-Silverman learning style theory. Interactive Learning Environments, 17(2),135-150.

P. Fissilis, A. Kameas, P. Pintelas. (1996). ORIENTMAN: an intelligent tutor for the ORIENT software development methodology. Software Engineering Journal, Julio.

Ford, N., & Chen, S.Y. (2000). Individual differences, hypermedia navigation, and learning: An empirical study. Journal of Educational Multimedia and Hypermedia, 9(4), 281-311.

Frias-Martinez, E., Chen, S. Y., & Liu, X. (2008). Investigation of behavior and perception of digital library users: A cognitive style perspective. International Journal of Information Management, 28(5), 355-365.

Gerjets, P., Scheiter, K., Opfermann, M., Hesse, F. W., & Eysink, T. H. S. (2009). Learning with hypermedia: The influence of representational formats and different levels of student control on performance and learning behavior. Computers in Human Behavior, 25(2), 360-370. doi: 10.1016/j.chb.2008.12.015

Gertner, A. S. and VanLehn, K. 2000. ANDES: A Coached Problem-Solving Environment for Physics, in Proc. ITS'2000, 133-142.

Gilbert, J.E., & Han, C.Y. (1999). Adapting instruction in search of "A Significant Difference. "Journal of Network and Computing Applications, 22(3), 149–160.

Graesser, C., Wiemer-Hastings, K., Wiemer- Hastings, P., Kreuz, R. and Tutoring Research Group. (1999). Autotutor: A simulation of a human tutor. Journal of Cognitive Systems Research, 1:35–51.

Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., and Harter, D. 2001. Intelligent Tutoring Systems with Conversational Dialogue, AI Magazine, 22(4):39-51.

Graf, S., Lin, T., & Kinshuk (2007). The relationship between learning styles and cognitive traits-Getting additional information for improving student modelling. Computers in Human Behavior, 24(2), 122-137.

Grigoriadou, M., Papanikolaou, K., Kornilakis, H., Magoulas, G. INSPIRE: An

INtelligent System for Personalized Instruction in a Remote Environment. In Proceedings of 3rd Workshop on Adaptive Hypertext and Hypermedia (Sonthofen, Germany, 2001), pp. 13-24.

http://wwwis.win.tue.nl/ah2001/papers/papanikolaou.pdf.

Honey, P. & Mumford, A. (1992). The manual of learning styles. Maidenhead, UK: Peter Honey Publications.

Hsieh, S. W., Jang, Y. R., Hwang, G. J., & Chen, N. S. (2011). Effects of teaching and learning styles on students' reflection levels for ubiquitous learning. Computers & Education, 57(1), 1194-1201.

Huang, C.-K. and Lin, C.-Y. (2011), Enhancing classroom interactivity and engagement: CFL Learners' perceptions of the application of Web 2.0 technology. British Journal of Educational Technology, 42: E141–E144. doi: 10.1111/j.1467-8535.2011.01219.x

Huang, E.Y., Lin S.W., & Huang, T.K. (2012). What type of learning style leads to online participation in the mixed-mode e- learning environment? A study of software usage instruction. Computers & Education, 58(1), 338–349.

Hwang, G. J., Kuo, F. R., Yin, P. Y., & Chuang, K. H. (2010). A heuristic algorithm for planning personalized learning paths for context-aware ubiquitous learning. Computers & Education, 54(2), 404-415.

Jonassen, D. H., & Grabowski, B. L. (1993). Handbook of individual difference, learning, and instruction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Koedinger, K. R., Anderson, J. R., Hadley, W. H. & Mark, M. A. (1997) Intelligent tutoring goes to school in the big city. Int. J. AI in Education, 8 (1), pp. 30-43.

Kolb, D. A. (1984). Experiential learning: Experience as the source of learning and development. Englewood Cliffs, NJ: Prentice-Hall.

Kubeš, T., (2007).Application of Adaptive Hypermedia Systems in E-Learning. Master's thesis CTU Prague, Czech Republic. http://www.tomaskubes.net/eat/.

Kulik, C.-L. and Kulik, J. 1991. Effectiveness of Computer-Based Instruction: an Updated Analysis, Computers in Human Behavior 7:75-94.

Kuljis, J., & Liu, F. (2005). A comparison of learning style theories on the suitability for elearning. In M. H. Hamza (Ed.), Proceedings of the IASTED Conference on Web-Technologies, Applications, and Services (pp. 191–197). Calgary, Canada: ACTA Press

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Latent Semantic Analysis passes the test: knowledge representation and multiple-choice testing. Unpublished manuscript.

Landauer, T. K., Laham, D., & Foltz, P. W. (1998). Computer-based grading of the conceptual content of essays. Unpublished manuscript.

Leen-Kiat Soh, Todd Blank(2008). Integrating Case-Based Reasoning and Meta-Learning for a Self-Improving Intelligent Tutoring System. International Journal of Artificial Interlligencein Education18,27-58.

Leen-Kiat Soh, L.D. Miller, Suzette Person ,Blank, T. (2005). ILMDA: Intelligent Learning Materials Delivery Agents, Computer Science & Engineering, University of Nebraska, Lincoln, NE.

Lester, J., Voerman, J., Towns, S., & Callaway, C. (1997). Cosmo: A life-like animated pedagogical agent with deictic believability. Working Notes of the International Joint Conference on Artificial Intelligence Workshop on Animated Interface Agents: Making Them Intelligent, 61–69.

Liegle, J. O., and Janicki, T. N., (2006). The Effect of Learning Styles on The Navigation Needs of Web-based Learners Computers in Human Behavior, pp. 885–898

Lin, Y. S., Hwang, G. J., & Kuo, F. R. (2009, November). Effects on cognitive styles in student achievement for context-aware ubiquitous learning. Paper presented at 17th International Conference on Computers in Education (ICCE 2009), Hong Kong.

LK Soh (2007). Integrated Introspective Case-Based Reasoning for Intelligent Tutoring System. Computer Science & Engineering, University of Nebraska, Lincoln, NE.

Lo, J. J., Chan, Y. C., & Yeh, S. W. (2012). Designing an adaptive Web-based learning system based on students' cognitive styles identified online. Computers & Education, 58(1), 209-222.

Mampadi, F., Chen, S. Y. H., Ghinea, G., & Chen, M. P. (2011). Design of adaptive hypermedia learning systems: A cognitive style approach. Computers & Education, 56(4), 1003-1011.

Melis, E., Andres, E., Büdenberder, J., Frishauf, A., Goguadse, G., Libbrecht, P., Pollet, M., and Ullrich, C. (2001) A generic and adaptive Web-based learning environment. International Journal of Artificial Intelligence in Education 12

Mitrovic, A. (2003) Supporting Self-Explanation in a Data Normalization Tutor. In: V. Aleven, U. Hopppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo, K. Yacef (eds) Supplementary proceedings, AIED 2003, pp. 565-577.

Moos, D. C. (2009). Note-taking while learning hypermedia: Cognitive and motivational considerations. Computers in Human Behavior, 25(5), 1120-1128.

MurrayTuroff. Designing a virtual classroom. In Inuited Papers /or the 1995 International Conference on Cornputer Assisted Instruction, March 1995.

Mustafa, Y. E. A & Sharif. S. M. (2011, Jan.). An approach to Adaptive ELearning Hypermedia System based on Learning Styles (AEHS-LS): Implementation and evaluation. International Journal of Library and Information Science. [Online]. Vol.

3(1), pp. 15-28. Available: http://www. academicjournals.org/ijlis

Paas, F., Tuovinen, J. E., van Merriënboer, J. J. G., & Darabi, A. A.(2005). A motivational perspective on the relation between mental effort and performance: optimizing learner involvement in instruction. Educational Technology Research & Development, 53(3), 25-34.

Papanikolaou, K. A., Mabbott, A., Bull, S., & Grigoriadou, M. (2006). Designing learner-controlled educational interactions based on learning/cognitive style and learner behaviour. Interacting with Computers, 18, 356–384.

Reiff, J. C. (1992). Learning styles. Washington, DC: National Education Association of the United States.

Riding, R.J. & Rayner, S. (1998). Cognitive Style and Learning Strategies: Understanding Style Differences in Learning & Behaviour. London, UK: David Fulton Publishers.

Ryokai, K., Vaucelle, C., and Cassell, J. (2002) "Literacy Learning by Storytelling with a Virtual Peer." In Proceedings of Computer Support for Collaborative Learning 2002.

García, P., Schiaffino, S. and Amandi, A. (2008), An enhanced Bayesian model to detect students' learning styles in Web-based courses. Journal of Computer Assisted Learning, 24: 305–315. doi: 10.1111/j.1365-2729.2007.00262.x

Sleeman, D., & Brown, J. S. (1982). Introduction: Intelligent Tutoring Systems. Intelligent TutoringSystems, D. Sleeman, J. S. Brown, Ed. Academic Press, 1982, pp. 1- 11.

Sternberg, R. J., & Grigorenko, E. L. (1997). Are cognitive styles still in style? American Psychologist, 52(7), 700-712.

Suppes, P., and Groen, G. (1967). M. Scandura (Ed.): Research in Mathematics Education. National Council of Teachers of Mathematics, 35-43.

Triantafillou, E., Pomportsis, A., & Demetriadis, S. (2003). The design and the formative evaluation of an adaptive educational system based on cognitive style. Computers & Education, 41, 87-103.

Triantafillou, E., Pomportis, A., & Georgiadou, E. (2002). AES-CS: Adaptive educational system based on cognitive styles. In P. Brusilovsky, N. Henze & E. Millán (Eds.), Proceedings of Workshop on Adaptive Systems for Web-Based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 1-11). Málaga, Spain: University of Málaga.

Tseng, J. C. R., Chu, H. C., Hwang, G. J., & Tsai, C. C. (2008). Development of an adaptive learning system with two sources of personalization information. Computers & Education, 51(2), 776-786.

van Seters, J. R., Ossevoort, M. A., Tramper, J., & Goedhart, M. J (2012). The influence of student characteristics on the use of adaptive e-learning material. Computers & Education, 58, 942-952.

Wang, T. & Mitrovic, A. (2002) Using neural networks to predict student's behaviour. In: Kinshuk, R. Lewis, K. Akahori, R. Kemp, T. Okamoto, L. Henderson and C-H Lee (eds.) IProc. Int. Conference on Computers in Education ICCE 2002, Los Alamitos, CA: IEEE Computer Society, pp. 969-973.

Wang, S. L., & Wu, C. Y. (2011). Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. Expert Systems with Applications, 38, 10831–10838.

Weber, G., & P. Brusilovsky (2001). ELM-ART: an adaptive versatile system for web-based instruction. International Journal of Artificial Intelligence in Education, 12, 351-384. http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html

Weller, H. G., Repman, J, & Rooze, G. E. (1995). The relationship of learning, behavior, and cognitive style in hypermedia based instruction: Implications for design of HBI. Computers in the Schools, 10(3/4), 401-420.

Witkin, H. A., Moore, C. A., Goodenough, D. R., & Cox, P. W. (1977). Field-dependent and field-independent cognitive styles and their educational implications. Reviews of Educational Research, 47, 1-64.

Wolf, C. (2003). iWeaver: Towards 'Learning Style'-based e-Learning in Computer Science Education. In Proc. Fifth Australasian Computing Education Conference (ACE2003), Adelaide, Australia. CRPIT, 20. Greening, T. and Lister, R., Eds. ACS. 273-279.

Woods, P. and Hartley, J. R. (1971), Some Learning Models For Arithmetic Tasks and Their Use In Computer Based Learning. British Journal of Educational Psychology, 41: 38–48. doi: 10.1111/j.2044-8279.1971.tb00656.x.

Zakaria, M. R., Moore, A., Stewart, C. D. and Brailsford, T. J., 2003. "Pluggable" User Models for Adaptive Hypermedia in Education 14th ACM Conference on Hypertext and Hypermedia (HT 2003), Nottingham, UK 26-30 August 2003.. CONF 14, 170-171.

Curriculum Vitae

Dehui Li

[redacted]

[redacted]

Specific Research interests:

Human Computer Interactions

Computer Science Education

Educational Psychology

Learning Sciences

Machine Learning

Artificial Intelligence

Specific Teaching Interests:

Introductory Programming

Human Computer Interactions

Database Management Systems

Web Development

Education:

Doctor of Science program in information technology

Jan.2013-expect December 2016

Towson University, Towson, Maryland

Dissertation topic: An Intelligent and Effective E-learning System That Provides Tailored Lessons to Students

Dissertation Committee: Dr. Song, Yeong-tae, Dr. Feng, Jinjuan, and Dr. Wang, Yuanqiong (Kathy)

Cumulative GPA of 3.7/4.00

Master of Computer Science

December 2012

Towson University, Towson, Maryland

Cumulative GPA of 3.82/4.00

Master of Public Administration

December 2008

University of Baltimore, Baltimore, Maryland

Cumulative GPA of 3.72/4.00

Bachelor of Business Management

December 2003

Beijing International Business Institute, Beijing, China

Cumulative GPA of 3.75/4.00

Research and Professional Experience:

Acclaim Systems, Harrisburg, PA (Software Engineer)

Jan.2015-Present

WCF Web services

Involved in CRM application development.

Involved in Cash management Project

Involved in Dynamic Form Project

Involved in Scanning Project on Windows Form Application

Involved in Pennsylvania State Election Project

Optimize database consistency, query and store procedures to improve the performance of project

Towson university, Doctor program Research

Jan.2013- Dec 2016

Involved in E-Learning Project using Python, SQLite and MS Sql

Involved in Text mingling and Sentiment analysis based on different classification algorism on NTLK

Involved in creating Learner's Profile Management, where the data is imported and exported in XML format to the Database.

Artificial intelligence on Adaptive E-learning

Graduate Assistant

Jan.2011-Jan.2015

Designed and implemented web application for College of Liberal Arts

Used .Net Validation Controls for server side validations

Developed the web pages using ASP.NET

Troubleshot and resolved bugs in .NET applications to ensure optimal development environment.
Design and develop Stored Procedures, DB Objects (Tables, Views) in SQL.
Involved with load and performance testing

Fashion electronics, Rockville, Maryland(Web Developer)

Feb.2014-Jan.2015

Technical maintenance

Develop E-commerce based website

E-marketing Management

Online store inventory management

Xcision LLC Intern, Columbia, Maryland

July.2012- July.2013

Developed and prototyped features and interfaces to meet business requirements

Involved in analyzing, modeling and developing front end (GUI) layer of the application using HTML, CSS.

Responsible for putting validations on various fields to avoid wrong or null data entry

Worked at ASP.Net platform using C# language for implementing business logic

Responsible for unit testing of the services as standalone application

Conference publications:

Li, Dehui, and H. Harry Zhou. "An Adaptive and Intelligent Tutoring System with Fuzzy Reasoning Capabilities." Journal of Convergence Information Technology 10, no. 1 (2015): 1.

Li, Dehui, and H. Harry Zhou. "An Intelligent Tutoring System with an Automated Knowledge Acquisition Mechanism." In Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on, pp. 88-91. IEEE, 2015.

Technical Skills:

DATABASES	MySQL, SQL, SQLite, Microsoft Access.
LANGUAGE	JAVA, C, C++, C#, VB, Ruby on Rails, Python
WEB LANGUAGES	HTML, XML, CSS, XSL, JavaScript
OPERATING SYSTEMS	UNIX, Windows 7, 2000, Windows 8
VERSION CONTROL	SVN, TFS, Git
REPORT	Microsoft SSRS
IDE	Eclipse, VISUAL STUDIO 2008, 2010, 2012 ,2015, Net Beans