UML-ALF AGENT BASED ADAPTIVE LEARNING FRAMEWORK: A CASE STUDY ON UML

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EFE CEM KOCABAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2010

Approval of the thesis:

**UML-ALF AGENT BASED ADAPTIVE LEARNING FRAMEWORK:
A CASE STUDY ON UML**

submitted by **EFE CEM KOCABAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ________________
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı ________________
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Veysi İşler ________________
Supervisor, **Computer Engineering Dept., METU**

Assoc. Prof. Dr. Kürşat Çağıltay ________________
Co-Supervisor, **Computer Education and Instructional Technology Dept., METU**

Examining Committee Members:

Prof. Dr. Petek Aşkar ________________
Computer Education and Instructional Technology, Hacettepe University

Assoc. Prof. Dr. Veysi İşler ________________
Computer Engineering Dept., METU

Assoc. Prof. Halit Oğuztüzün ________________
Computer Engineering Dept., METU

Dr. Ayşenur Birtürk ________________
Computer Engineering Dept., METU

Asst. Prof. Dr. Saniye Tuğba Bulu ________________
Computer Education and Instructional Technology, METU

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name	: Efe Cem KOCABAŞ
Signature	:

ABSTRACT

UML-ALF AGENT BASED ADAPTIVE LEARNING FRAMEWORK:
A CASE STUDY ON UML

Kocabaş, Efe Cem

M. Sc., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Veysi İşler
Co-Supervisor : Assoc. Prof. Dr. Kürşat Çağıltay

July 2010, 95 pages

As the amount of accessible and shareable knowledge increases, it is figured out that learning platforms offering the same context and learning path to all users can not meet the demands of learners. This issue brings out the necessity of designing and developing adaptive hypermedia systems. This study describes an agent-based adaptive learning framework whose goal is to implement effective tutoring system with the help of Artificial Intelligence (AI) techniques and cognitive didactic methods into Adaptive Educational Hypermedia Systems (AEHS) in the domain of Unified Modeling Language (UML). There are three main goals of this study. First goal is to explore how supportive agents affect student's learning achievement in distance learning. Second goal is to examine the interaction between supportive agents and learners with the help of experiments in Human Computer Interaction laboratories and system analysis. The effects of the methodology that agents give misleading hints which are common mistakes of other learners are also investigated. Last goal is to deliver effective feedback to students both from IAs and tutors.

In order to assess that UML-ALF has accomplished its objectives, we followed an experimental procedure. Experimental groups have taken the advantage of adaptive and intelligent techniques of the UML-ALF and control groups have used the traditional learning techniques. The results show that there is a positive correlation between variables practice score and number of agent suggestion which means, as the participants benefit from supportive agents, they get higher scores.

Keywords: Adaptive Learning, Intelligent Tutoring Systems, Intelligent Agents, UML, Distance education.

ÖZ

UML-ALF ETMEN TABANLI UYARLANABİLİR ÖĞRENME SİSTEMİ:
UML ÜZERİNE BİR ÇALIŞMA

Kocabaş, Efe Cem

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Veysi İşler
Ortak Tez Yöneticisi : Doç. Dr. Kürşat Çağıltay

Temmuz 2010, 95 sayfa

Erişilebilen ve paylaşılabilen bilginin miktarı arttıkça, aynı içerik ve eğitim yolunu sunan eğitim platformlarının tüm kullanıcıların isteklerini karşılayamadığı anlaşılmıştır. Bu durum, uyarlanabilir eğitsel hiper ortamların tasarlanmasının ve geliştirilmesinin gerekliliğini ortaya çıkarmıştır. Bu çalışmada, Birleşik Modelleme Dili (UML) alanında yapay zeka teknikleri ve kavramsal eğitim metodlarının yardımıyla etkili bir öğretim sisteminin uygulamaya döküldüğü etmen tabanlı bir uyarlanabilir öğrenme sistemi tanımlanmıştır. Bu araştırmanın üç ana hedefi bulunmaktadır. Birincisi, destekleyici etmenlerin öğrencilerinin başarısını nasıl etkilediğini araştırmaktır. İkinci hedef, destekleyici etmenler ve öğrenciler arasındaki etkileşimi İnsan Bilgisayar Etkileşimi Laboratuvarı ve sistem analizleri yardımıyla incelemektir. Akıllı etmenlerin, diğer kullanıcıların ortak hatalarından elde ettiği yanıltıcı ipuçlarını kullandığı öğretim tekniğini de bu kapsamda araştırılmıştır. Son hedef ise, öğrencilere akıllı etmenler ve eğitmenler tarafından verimli geri dönüşler yapabilmektir. UML-ALF çalışmasının hedefine başarıyla ulaştığını belirlemek için deneysel bir prosedürün takip edilmesine karar verilmiştir. Deneysel grup, sistemin uyarlanabilir ve akıllı tekniklerinin tümünden faydalanırken, kontrol grubu geleneksel öğrenim tekniklerini kullanmaktadır. Sonuçlar kullanıcı puanı ile akıllı etmen önerileri arasında pozitif bir korelasyon olduğunu, yani katılımcı akıllı etmenlerden faydalandıkça puanının arttığını göstermiştir.

Anahtar Kelimeler: Uyarlanabilir Öğrenme, Akıllı Öğrenme Sistemleri, Akıllı Etmenler, UML, Uzaktan Eğitim.

To my parents

and

to my beloved

ACKNOWLEDGMENTS

Numerous individuals provided support, guidance, and encouragement during my study as a graduate student, and it is only appropriate that I acknowledge their valuable contributions.

First of all, I would like to extend my deepest gratitude to my thesis advisor Assoc. Prof. Dr. Veysi İşler for all of his enduring support, tireless revision, and encouragement. I would like to express sincere gratitude to my co-advisor Assoc. Prof. Dr. Kürşat Çağıltay for his valuable suggestions and comments.

I would like to express gratitude to my examination committee members.

I would like to express thanks to Assoc. Prof. Dr. Kürşat Çağıltay and other members of Human Computer Interaction Research and Application Laboratory of METU Computer Center who provides support for the study.

I would like to thank Murat Miran Köksal and Didem Tufam Avdan for their encouragement and support.

I would like to express gratitude to my colleagues and friends for their encouragement and support. I would also like to thank to all participants who joined in experiments of the study.

Lastly, I would like to express my love and deep gratitude to all members of my family and my girlfriend, İlkay Kayserilioğlu, for their patience, encouragement and for their steady trust in me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

ABBREVIATIONS

AA: Adaptive Agent

AEHS: Adaptive Educational Hypermedia Systems

AI: Artificial Intelligence

AICC: Aviation Industry CBT (Computer-Based Training) Committee

AOI: Area of Interest

ARP: Automated Role Players

BSc: Bachelor of Science

BTN: Behavior Transition Networks

CAI: Computer Aided Instruction

DA: Diagnostic Agent

FM: Feedback Mechanism

GPS: Global Positioning System

HACP: HTTP AICC Communication Protocol

HCI: Human Computer Interaction

IA: Intelligent Agent

IMS: Instructional Management Systems

ITS: Intelligent Tutoring Systems

LA: Language Agent

LMS: Learning Management System

MA: Model Agent

MCQ: Multiple Choice Questions

PEAS: Performance Measure - Environment - Actuators – Sensors

PSI: Personalized System of Instruction

SA: Supportive Agent

SCORM: Shareable Content Object Reference Model

SME: Subject-matter Experts

TAO: Tactical Action Officers

UML: Unified Modeling Language

UML-ALF: UML - Adaptive Learning Framework

XML: Extensible Markup Language

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

With the development of network technology and disadvantages of traditional education, distance education has started to take an important role for method of learning. Providing an effective distance education system becomes a hot research topic in the last decade. Motivation is also a significant factor for an effective learning. Learners should take an active role in their learning so that they are kept motivated. As the amount of accessible and shareable knowledge increases, it is figured out that learning platforms offering the same context and learning path to all users can not meet the demands of learners (Brusilovsky, 2001). This issue brings out the necessity of designing and developing adaptive hypermedia systems. The main principle of adaptive learning is that no single didactic strategy is good for all learners (Papadimitriou, Grigoriadou & Gyftodimos, 2009). As a result, learners will be well-concentrated and well-motivated while they are dealing with their learning goals when educational procedures are adapted to their user model. Each learner may have different learning styles which expose the characteristic differences. Therefore each learner reacts different to various educational approaches and methods (Kolb, 1984). There are many researches that analyze learning styles and preferences of the learners in order to adapt educational approaches according to their preferences (as cited in Papadimitriou et al. 2009)

1.1.1 Adaptive Educational Hypermedia Systems (AEHS)

Adaptive Educational Hypermedia Systems (AEHS) are considered to be the solution to the problems of traditional e-learning systems. In such traditional systems 'one-size-fit-all' approach is used whereas in AEHS it is personalized based on user's abilities, characteristics and knowledge. AEHS also provide solution to the 'lost in hyperspace' syndrome, in which user get confused due to the complex interface of hypermedia. Several adaptive and intelligent techniques have been applied to AEHS so far (Papadimitriou et al., 2009; Brusilovsky & Peylo, 2003; Brusilovsky, 2000). Adaptive techniques can be listed as:

Adaptive Presentation: Adaptive Presentation adapts the learning material according to user's background, characteristics and knowledge of the user model.

Adaptive Navigation Support: The purpose of this method is to guide user in hyperspace due to the user model features. The most common ways of adaptive navigation support are direct guidance, link sorting, link annotation, link hiding, disabling, and removal (Brusilovsky, 2000)

Adaptive Collaboration Support: Mainly the goal of this technique is to match different students in a group according to their characteristics. However, in this work the purpose is to use IAs as co-workers and explore the effect of collaboration.

Intelligent techniques can be classified as follows:

Curriculum Sequencing: one of the oldest ITS technologies. Its purpose is to present user a learning path whose learning material is carefully selected for the user.

Intelligent Analysis of Student Solutions: After user submits the answer a solution analyzer decide whether the result is correct or not. Then, feedback mechanism takes the initiative and returns the response.

Interactive Problem Solving Support: Without waiting for the final answer of the user helping during each step of the problem.

Example-Based Problem Solving Support: Its purpose is to help users solving the problem by leading them to the related and successful solutions which they experienced before.

1.1.2 Intelligent Tutoring Systems (ITS)

Intelligent Tutoring Systems derived from Artificial Intelligence (AI). At the same time AI was dealing with the idea of creating machines that can think like humans. As researchers came across to the problems that were difficult to solve, they realized that trying to imitate human cognition with computers needless, since human beings do not think like computers. Thus, researchers headed to another research area like expert systems which was more productive.

Intelligent tutoring systems consist of four different components: the interface module, the expert module, the student module and the tutor module. The interface module is the module where the interaction of the learner with ITS is provided. Generally interface module is formed by a graphical user interface and it is sometimes a simulation environment for a specific domain. Expert module is an agent-based intelligent environment including knowledge-based description that ITS is teaching. Student module includes the information of student's knowledge level and individualized preferences. Tutor module tries to take place of the human tutor in ITS. It supports learners by providing feedback and remedial instruction. (Feng, Heffernan, N.T., Heffernan, C. & Mani, 2009)

In the 1960s, researchers created a model called computer-aided instruction (CAI) (Uhr, 1969). This model is basically designed to examine the learner with a problem, receive and record the learner's response. Thus, it determines the achievement of the learner. However, this system only assumes that when it produces learning material, learner would acquire information easily. But, it does not find a solution to the way how people learn. An intelligent tutoring system (ITS) is an advanced version of the CAI model. ITS is any computer-based learning system that teaches learning materials and supports students without the contribution of human tutors. The main principle applied in ITS is the theory of learning by doing.

All along the time, artificial intelligence was an important area for ITS research. There are too many learning systems which were built around expert systems. However, another field of artificial intelligence identified as agents and multi-agent systems has affected ITS research recently. Most conferences now focus on pedagogical agents (Mengelle, de Léan & Frasson 1998). There are many reasons for coming out of this new field. Agents have advantages of modularity and flexibility with the attitude of computer science. There are several works studying knowledge gaining and learning skills for agents up to now such as (Huffman, 1994), (Tecuci & Hieb, 1996). Agents might be described as intelligent collaborators helping learners in achieving specific goals. They can make proper actions and response to the change of the environment without the participation of users.

Russell and Norvig (2003, p. 32) defined an agent as ‘anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators’. Agents can be categorized as the way they perceive the environment, including human agent, robotic agent and software agent. A human agent uses its sense organs such as vision, hearing, smell, taste, and touch for sensors and hands, legs, mouth and other body parts for actuators. A robotic agent might have cameras, thermometers and GPS as sensors and various motors for actuators. For a software agent, any input such as knowledge database, file structures, network packets, keystrokes or mouse clicks might be its sensors whereas any output such as writing files, printing to the screen might be its actuators. In this study, we focus on software agents.

The agent function for an agent maps every possible percept sequence to a possible action that agent can take. The agent program implements the agent function for an artificial agent. A proper design for an agent program is contingent upon the nature of the environment. A performance measure describes the standards for a successful agent behavior. It evaluates the performance of an agent in an environment. Clearly, there is no such performance measure suitable for all agents. An agent tries to maximize the expected value for this evaluation. For this reason, a performance measure must be selected carefully. Agents are grouped into five classes based on their degree of perceived intelligence and capability: (Russell & Norvig, 2003) simple reflex agents, model-based reflex agents, goal-based agents, utility-based agents and learning agents. It is also accepted that all agents can improve its performance by learning.

1.1.3 Learning Management System

Although there is no universally accepted concept of the information society, it is believed that it is started somewhere between the 1970s. While the total amount of information doubled on each century till 1850s, this period decreased to 5 years in 1970s. From 1980s, renewal of the information decreased to 1 year. Thus, total produced information in last 30 years is greater than previous 5000 years (Somyürek, 2009). In such a time that information is so intense and changing rapidly, it become obvious that business and academic world need adaptive learning environments. Most companies and universities teach their employees and students with Learning Management Systems (LMSs).

LMS is a common term generally used for large systems that manage online learning events for students, instructors, and administrators. These systems mostly provide e-learning programs, specific contents, training modules, and communication tools for their participants. The most known open source LMSs are Moodle and Sakai, commercial LMSs are WebCT and Blackboard.

E-learning, which is becoming widespread day by day and is being laid as infrastructure by most of the universities, has both pros and cons. It is a student-centered environment. A student may study a subject until learning it. There is no time limit. Student may communicate with instructors and other students with various communication tools to understand the topic and benefit from the experience of someone all over the world. Therefore the response time for reaching the desired resource is very little. Cost of education is also decreased. It is independent from time and space. Course materials can be reorganized due to daily circumstances. Student can try himself with several exams and tests. Rapid feedback can provide high motivation. All e-learning activities of someone can be reported. However, instructors should know how to provide an efficient e-learning. It is difficult for them to transfer traditional course material into online world. The requirements of e-learning facilities and technical education and support for students and instructors are the other issues to be considered. There are different e-learning types. They can be classified as web based instruction, synchronous and asynchronous instruction, virtual education, computer based distance education, internet based/aided education and online education. However, traditional hypermedia systems such as LMSs that provide same page contents and navigation links are inadequate in meeting the requirements of learners interested in personalized learning desires. Demand of each learner differentiates due to their characteristics, different learning styles, different information processing styles and choice of using different knowledge resources. Adaptive hypermedia systems create a user model for each user considering their various features and offer personalized settings. A system must provide some circumstances in order to be an adaptive one. They need to have a hypermedia system, this system ought to contain user modeling and system has to be adapted to the user, using this user model. Adaptive learning systems are also closely related with Intelligent Tutoring Systems and Artificial Intelligence. A learning content is adapted to each user and a learning path is dynamically guided with the help of intelligent tutoring systems. While adapting the information and its presentation, adaptive learning system uses artificial intelligence. Main goal of adaptive hypermedia systems is to provide individualization of the system according to specific needs and preferences of each learner.

1.2 Purpose of the Study

This study describes the UML-ALF – UML Adaptive Learning Framework – whose goal is to implement effective tutoring system with the help of AI techniques and cognitive didactic methods into Adaptive Educational Hypermedia Systems (AEHS). There are three main goals of this study. First goal is to explore how collaborative learning – work affect student's motivation in distance learning. In this part, system uses artificial intelligence (AI) techniques to provide Intelligent Agents (IAs) which take supportive agents' place (Remolina, Ramachandran, Stottler & Davis, 2009). Second goal is to deliver effective feedback to students both from IAs and tutors. This is one of the research problems in the field of Intelligent Tutoring Systems (ITSs) (Fossati, Eugenio, Brown, Ohlsson, Cosejo &

Chen, 2009). Last goal is to present an effective AEHS which is adapted according to characteristics of learners in the domain of Computer Science about UML. Among the criticism about UML, especially interested in one of them – "Problems in learning and adopting make UML problematic." (Bell, 2004) – In this study, adaptive techniques and intelligent techniques such as Adaptive Presentation, Adaptive Navigation Support, Curriculum Sequencing, Intelligent Analysis of Student Solutions and Interactive Problem Solving were used.

1.3 Research Questions

The following research questions are posed, in order to achieve the purpose of the study:

1. Do supportive agents provide contribution to the learning achievement of learners in distance education?
2. Are learners interacted with supportive agents?
   2.1 Does the methodology that agents give misleading hints which are common mistakes of other learners have positive effects on learning the subject.
   2.2 Is the methodology which forces learners to renew their knowledge as they make mistakes effective on their learning achievement?
3. Can intelligent agents and computer based tutors give effective feedback to the learners which is close to the performance of human tutors?

1.4 Significance of the Study

As the amount of accessible and shareable knowledge increases, it is figured out that learning platforms offering the same context and learning path to all users can not meet the demands of learners (Brusilovsky, 2001). Also, the contribution that is provided by tutors in classroom education does not exist in distance education. These issues bring out the necessity of designing and developing adaptive hypermedia systems. It is expected that this study determines how adaptive hypermedia systems and intelligent agents contribute on distance education. Besides, this study gathers the information of which methodologies of adaptive and intelligent techniques have positive effects on learning, which learning path the participants follow, how they react to the learning system, with the help of analysing data of intelligent agents, face-to-face trials and observations of experiments in Human Computer Interaction Labs.

1.5 Assumptions

It is assumed that the participants of the study responded the questions properly, accurately, and objectively. Moreover, it is accepted that language capability of the participants are enough for learning materials.

1.6 Limitations of the Study

Adaptive part of this study was formed of unprofessional learning contents which are not designed by the experts of UML domain. Thus, learning gain of participants is limited to the success of these contents.

Because of the nature of distance education and time limitation, participants took learning materials on their own computers. Consequently, the validity of results in learning process is limited to the honesty of the participant behavior.

During data gathering process, HCI experiments were made with 5 participants due to the time and scope limitations.

1.7 Following chapters

In Chapter 2, related works which have been done so far are being investigated. Later on, in Chapter 3 proposed model of the UML-ALF is explained. User Modeling and Supportive Agent modeling, Adaptive Educational Hypermedia Systems, adaptive engine, feedback mechanism of the system and intelligent agents are represented in details. In Chapter 4, a case study of this study - UML - is introduced. Also, implementation of the web based hypermedia system is illustrated. In Chapter 5, evaluation of the system with detailed analysis, graphs and charts are discussed. Lastly, conclusion of the thesis and future work is stated in Chapter 6.

CHAPTER 2

RELATED WORK

This chapter is organized by using thematic review. The related works are grouped and discussed in terms of the themes or topics they cover. The purpose of preferring thematic review rather than chronological one is to demonstrate the types of topics which are important to this research. There exist three groups including Adaptive Learning Environments, Intelligent Tutoring Systems and Intelligent Adaptive Learning Environments. All groups use intelligent agents in their systems. However, in first group just adaptive techniques are used and similarly second group uses just intelligent techniques. Third group is the hybrid of other two groups where both adaptive and intelligent techniques are used.

2.1 Adaptive Learning Environments

Most LMSs have free-browsing learning mode. However, Hong, Chen, Chang & Chen (2007) declare the idea that no such learning path is convenient for all learners and argue that this mode decreases learner's performance due to inappropriate course selection or disorientation during learning period. Thus it is thought that deciding personalized syllabus is a significant research issue for e-learning systems.

According to the work done by Hong et al. (2007), an appropriate learning path for each individual is generated with the help of the result of pre-tests. In order to prove the quality of this method, experimentation is made with some primary school students who are dealing with the 'Fraction' subject. A learning path is generated according to the pre-tests. Then, two sample groups are created. First group is trained by free-browsing method. Second group used the learning path which is formed with the learning –path generation algorithm. After the post-test results the achievement of first group is about %32 and second group's success is almost %48.

Due to successful results of this research, learning paths of users in our thesis are determined with the results of pre-tests. Moreover, some questionnaires are used in order to take more information about user, especially for adaptive learning in our study. Thus, more effective learning paths are created.

Results can also decide the difficulty of the content and detail level. Experimental results show that the proposed method can precisely provide personalized curriculum sequencing based on difficulty parameter and concept continuity of successive courseware, and moreover can accelerate learner's learning effectiveness.

Cabukovski (2006) also presents the idea of offering different content views according to the level of each user. He estimates the level of the student with an agent based testing subsystem. This agent infrastructure is built on an existing distance education system called MATHEIS which is a domain of mathematics and informatics. System supports XML, SCORM and AICC standards. System has Supervisor Agent, User Agent, Level Agent and Expert Agent. Briefly, all agents work for determining the level of user and presenting the proper content to the learner. However, it is just one of the adaptive techniques - adaptive presentation - of Adaptive Agent in our study.

Jin & Ming (2008) presents an intelligent agent system whose purpose is to find out the interest of each user in different domains dynamically. They use Naive Bayesian Classifier approach for determining user's interests. It provides recommendation for newly created contents according to the inferences that has been previously made. System has three kinds of intelligent agents: Personal Agent, Tutor Agent and Information Agent. Unlike UML-ALF, this study works on multiple domains. Therefore, personal agent analyzes user interests and can recommend interesting materials from different domains. Tutor agent, helps users to find out proper materials when they ask for help and users notify information agent about their demands and get response from information agent according to their pre-specified demands. This study also aims to offer adaptive presentation for their users which is just one of the duties of adaptive agent of our framework. However, since there are not enough data, the accuracy of the system is stucked in 50%-60%.The advantage of this research is analyzing and recommending various materials on multiple domains. But in our study, a learning path is assigned for users adapting their user modeling and users are forced to follow this path. The advantage of our framework is providing various adaptive and intelligent techniques to the user during the whole process of learning.

Chen (2008) also uses intelligent agents for improving e-learning process by applying an adaptive process. It suggests using portfolios which is a method to present selfability in different branches of industry for adjusting learners' education. In this research intelligent agents are divided into two groups: Learning Assistants and Teaching Assistants. These groups may stand for Study and Practice interfaces in our study. Agent based system learning process starts with signing of the learner to the system. First User Agent collects the user's portfolio then sends the necessary information to Learning Agent in order to present adaptive learning material to the learner. Adaptive Agent which is responsible for the same duty in our framework communicates with User Model in order to present adaptive material.

2.2 Intelligent Tutoring Systems

In order to develop computer programs, mostly software design methodologies are used. Those methodologies are set of procedures which are followed during the development process. It is considered that it is a creative process and cannot be reduced to standard routines. However, it must have a design structure as well. Software design process mostly consists of those steps: requirements, design, implementation, module test, integration test, system test, acceptance and deployment.

Bowman, Lopez Jr., Donlon & Tecuci (2001) describe a learning environment that software design methodologies are taught by intelligent agents. The user specifies the requirements and the software development team, after gathering data and clarifying some ambiguities, selects and uses a design methodology to develop the desired software product. The process is capable of making an error. There is a difficult problems set and a group of people who are highly qualified and are able to solve such problems. These definitions are called as knowledge-based problems and subject-matter experts (SMEs).

Developing knowledge-based software agents that incorporate the expertise of the SMEs is a widespread problem for AI field researchers (Bowman et al., 2001). Agents consist of two main components: a knowledge base and an inference engine. The knowledge base contains the data structures representing the entities from the expert's application domain such as objects, relations between objects, classes of objects, laws, actions, processes, and procedures. The inference engine consists of the programs that manipulate the data structures in the knowledge base in order to solve problems in a way that is similar to how the expert solves them. Knowledge base of Supportive Agents is held in Agent Modeling and user knowledge base is defined with User Modeling in our framework.

In classical knowledge-based agent development process, there exists a contradiction. A knowledge engineer is both part of solution and problem. In addition to this, development of each new knowledge base or agent starts from the beginning, and it is a difficult process. The goal is to enable SME that has no initial knowledge to develop a knowledge base agent. This approach is called Disciple. It is based on developing learning agents which are capable of using knowledge representation that makes easier knowledge reuse and learning, and can be directly taught by an SME. Since each Supportive Agent and user knowledge base starts from scratch there is no need for reusing the information in UML-ALF domain.

There are several learning strategies such as learning from examples, learning from explanations, and learning by analogy, in order to acquire the knowledge from the SME, which are integrated by the learning and knowledge acquisition engine (Bowman et al., 2001).

Knowledge base consists of two main components: object ontology and a set of reduction and composition rules. Reduction rule is an if-then structure which expresses the simpler problems that are reduced from the main problem. A composition rule is also an if-then structure which combines the simpler solutions into a general solution of the problem that is stated in reduction rule. Supportive Agents in UML-ALF also learn with reduction and composition rules. Under this agent-building paradigm, knowledge engineers customize the graphical user interface, help the SMEs to learn how to teach the Disciple agent and make the reuse of ontological knowledge easier to support SME's creation of agents. This paradigm eliminates much of the error generated by the many different people involved in the typical framework for software development.

Remolina et al. (2009) introduce the study "Rehearsing Naval Tactical Situations Using Simulated Teammates and an Automated Tutor" which is about developing a simulation-based Intelligent Tutoring System for training Tactical Action Officers (TAOs). TAO is the responsible officer in command center in navy ship of the entire watch team. ITS' role is to implement the same environment for the TAO to supervise. It uses Artificial Intelligence algorithms to provide Automated Role Players (ARP) which are actually watch standers that inform the TAO about the events in real environment.

Remolina et al. (2009) provide the communication between TAO trainees and ARPs through a speech recognition unit, which understand TAO trainees' commands in natural language and transfer them to text and vice versa. ARPs are the equivalent of SAs in UML-ALF. However, the communication between learner and SAs is provided by Language and Diagnostic agents. The system is learn by doing tactical decision making ITS. Trainee interacts with simulated teammates using a natural language interface. In order to simulate the behavior of AI agents Behavior Transition Networks (BTNs) are used. It is accepted that the most risky part of the project is the natural language interface, since it is based on trainee's speech output.

According to the grammar syntax of recognition system, user has to choose correct commands. Before the advent of the system an instructor is needed for every two students, but after this system now an instructor is needed for 42 students. Since BTNs are peculiar to domain of this study, we preferred knowledge base decision support with simple reflex, model and goal based agents in our study.

Similarly, Fossati et al. (2009) develop an ITS system which helps students to understand the subject of Linked Lists easily. It is an important topic in Computer Science. It is believed that most of the computer science students have difficulties in learning this subject. According to the feedback of students, system is considered interesting, useful and its performance is getting closer to the performance of human tutors.

Fossati et al. (2009) state that iList project is an interdisciplinary research whose purpose is to understand the effective tutoring methods. It is believed that iList project has reached some maturity that it can be used in computer science courses. According to the results of this research, the importance of positive feedback is understood. iList is now delivering mostly negative feedback to the mistakes of students. In order to generate more feedback it is decided to monitor student actions more closely. UML-ALF delivers both positive and negative feedbacks, with verbal and non-verbal responses. Feedback Mechanism (FM) is responsible for this task and Diagnostic Agent help FM to produce more effective feedbacks.

As well as Remolina's approach, Faria, Silva, Vale & Marques (2009) also study about developing a simulation-based Intelligent Tutoring System. Power Systems' performance is needed to be effective. Control Center operators are responsible for dealing with this task. Therefore, operators' actions may end up with incident such as power failure. The study is about developing an Intelligent Tutoring System (ITS) for training those Control Center operators.

Many artificial intelligence algorithms are introduced during the development of this system such as Multiagent Systems, Neural Networks, Constraint-based Modeling, Intelligent Planning, Knowledge Representation, Expert Systems, User Modeling, and Intelligent User Interfaces. It has two main tasks. One of them is Incident Analysis and the other one is Diagnosis and Service Restoration. The system is used by Electrical Engineering undergraduate students who will possibly be Control Center operator in the future. The system is used for two purposes. For training BSc students which will probably will be hired for job and for training already hired operators.

This environment is selected as one of the most important systems using AI algorithms in Portugal (Faria et al. 2009). System has a connection with SPARSE, which is an expert intelligent alarm processing system. It is also used for fault diagnosis training. It provides model tracing technique to monitor operator's reasoning. It either generates scenarios from real cases or creates new scenarios for trainees. The difficulty level of problems can be automatically assigned. With the help of Multiagent Systems paradigm, modeling the interaction of several operators during system renewal is possible. Constraint-based Modeling technique in restoration training can be used.

2.3 Intelligent Adaptive Learning Environments

Mengelle et al. (1998) use pedagogical intelligent agents inside Intelligent Tutoring Systems (ITS). Agents' purpose is to help students to learn using pedagogical strategies. In this manner, an agents and multi-agents system which is a field of artificial intelligence has influenced this research.

Mengelle et al. (1998) focus on three main points: the actor (agent) paradigm, communication techniques with the domain material and representation of this material. Actors are defined as reactive, instructable, adaptive and cognitive agents. They have two main features, react to the others and are able to learn. These agents are equivalent with supportive agents (SAs) in our study.

They define an interpreted language to help building actors. Student is faced to two types of agents: a teacher and a companion. When a student asks for help, companion dialogs with the resource which includes complete answer and a hint. Companion should decide whether to give a hint or complete answer. In our research learner only face with Supportive Agents which is called companion in this study. After SAs suggest their opinions, a scaffolding approach is used between tutor and learner in order to decide whether to give the complete answer or a hint that leads learner to the answer.

Mengelle et al. (1998) state that the communication protocol between actors and resources consists of four kinds of messages. Firstly, a given source is loaded with the basic information such as its type. Then, resource informs the actors each time, when student interacts with the resource. Resource may reply with several ways: reformulating question, giving a correct hint, giving a misleading hint, giving complete answer. At last step, actors decide what to do and choose one possible reaction due to their own expertise.

In order to be more effective, communication between the resource and supportive agents is provided with the help of Language Agent and Diagnostic Agent in architecture of our study. In addition to Mengelle, Dweck (2007), Izzo & Mazzone (2008) and Chialvo & Bak (1999) also express that learning from mistakes is one of the learning methodologies. Thus, in each practice some agents may give misleading hints which are common mistakes of other learners in UML-ALF. But learners are informed of this situation in order to be aware of the misleading hints. Also, tutor agent provides the correct hint using scaffolding approach after SA's suggestion.

While deciding the technique for building resources, the example of a classical tool for knowledge assessment -Multiple Choice Questions (MCQ) - has been taken (Mengelle et al., 1998). It allows straightforward diagnosis and quickly building of efficient resources. The resource consists of four main parts: identification part, subject part, answer processing part and a set of hints. Identification part is formed by an identification id and a list of related knowledge. Subject part describes the question text and choices. An answer processing part provides evaluating the knowledge of student and finally hints which may both help and mislead student, are involved in resource.

Question-ID	*<xxxx>*	
Related knowledge	*<capital-id>, <communism-id>*	
Question	Choose the towns that are or have been capital of communist countries: 1. Rome 2.Moscow 3. Stalingrad 4.Bucarest	
Refinement of the question	Null	
Refinement of the answers	1	Rome is a town located in Italy. It is well…
	3	Stalingrad is now named…

Answer	**Knowledge ok**	**Knowledge not ok**
1---	*<capital-id>*	*<communism-id>*
-2--	*<capital-id>, <communism-id>*	
(…)		

Hints	**T/F**	**Text**
<capital-id>	*true*	Generally speaking, capitals are important cities.
<communism-id>	*true*	Most of communist countries were in East Europe.
<communism-id>	*false*	The proportion of communist countries is the same in each continent.

Figure 2.1 A generic structure to implement MCQ manageable by actors

Similar resource structure is used in our framework. However, since domains of these studies are different, techniques for building the resource are also different. Language agent is responsible for building and interpreting the resource. The question designer links each possible answer with a both mastered and not-mastered knowledge.

Mengelle et al. improve actor knowledge expertise by two tasks: diagnosis and revision. The diagnosis task controls whether the behavior of the actor respect its goals. Each actor has two main goals: its individual goal and the collective goal which is to help learner. Each is goal is divided into sub goals. Diagnosis task checks every sub goal and find possible problems. Then revision task aims to fix those problems to improve actor's knowledge.

All these processes actualize in our study almost in same way. Supportive agent (SA) decides its answer according to its knowledge base while Diagnostic agent helps SA communicating with Agent Model and resource. Also Diagnostic agent evaluates SA's performance and updates its knowledge level in agent model.

Papadimitriou et al. (2009) introduce a new adaptive educational system called MATHEMA. The purpose of the system is to help high school students interactively by eliminating their misunderstandings and learning difficulties. The domain of the research is about electromagnetism in Physics.

Students may work in system both individually and collaboratively. Students may follow an activity which is formed of experimentations, simulations, explorations, etc. The experimental results show that following such an activity increase student's achievement and affect their motivation.

Adaptive Educational Hypermedia Systems (AEHS) can be a key solution to the traditional online education systems. The problems of traditional systems are static content, "lost in hypermedia" syndrome and 'one-size-fits-all' approach (Papadimitriou et al., 2009).

AEHS uses the ideas of Hypermedia and Intelligent Tutoring systems in order to fulfill the requirement of producing such applications which can be adapted to the each learner's knowledge level, background, learning goal, style and cognitive preferences. In order to introduce adaptation in web-based AEHS, several adaptive and intelligent techniques have been applied. These are; Curriculum sequencing, Adaptive presentation, Adaptive navigation support, Interactive problem solving support, Intelligent analysis of student solutions, Example-based problem solving support, Adaptive collaboration support—adaptive group formation and peer help. This study uses most of adaptive and intelligent techniques that our framework uses, but not all of them. According to the experiment results, none of the students answered all 12 questions correctly. After simulation each of them increased their correct results. Then, with the help of collaboration process and the dialog with the systems, at the end all of the students made 12 correct answers. An experimental study with senior high school students showed that the MATHEMA helps the students improve their performances. The evaluation techniques and results are so impressive that evaluation methodology of our study is originated from this research.

CHAPTER 3

PROPOSED MODEL

In this chapter a model of adaptive educational system with supportive agent infrastructure will be presented. This model offers an effective learning to the user which is adapted according to individual skills and preferences of user. Also it supports learner during his/her practice period with the help of agents using intelligent techniques. This model is formed by a framework that includes knowledge base models (User Model, Agent Model), data structures, agents (Adaptive Agent, Model Agent, Diagnostic Agent, Language Agent, Feedback Mechanism, and Supportive Agents) and their communication. Following sections analyzes the proposed model in details. A case study of this model is implemented on UML domain which is described in next chapter.

3.1 User Profiles

UML-ALF consists of three main profiles. User profile, Tutor profile and Admin profile. System admin can create other profiles. A new supportive agent is also created and its knowledge level, settings are determined by this profile. Evaluation of users and whole system can be monitored through the admin profile. Tutors can create course contents, exams and practices for learners. User profile is the main profile of the framework. Necessary information about user's knowledge level and preferences is gathered with the help of the pre-tests and questionnaires when user signs up to the system. User may follow the learning path which is assigned by the system or practice his/her knowledge in practice area.

3.2 User Modeling

While designing the User Model, modeling standards of IMS Learner Information

Packaging (IMS, 2001) and The Max Model (Lynch, Palmiter & Tilt, 1999) is based on.

3.2.1 User Model's User Profile:

User Profile is composed of two data structures as User Info and User Knowledge. User Info has two main components from IMS Learner Information Packaging (IMS LIP) - identification and security key -Identification stands for the learner information which contains attributes such as user codes, names, addresses, contact information, status, enrollment date, etc. Security key is used to keep the passwords and authentication keys of user.

User Knowledge data structure has four main components from IMS LIP which are relevant to this system. These components are Goal, Accessibility and Competency. Goal is the description of learner's personal objectives. It is defined as learning path of user in this system. Accessibility defines the cognitive, technical and physical preferences of the learner. Lastly, competency is the skills that learner gained from the learning system.

3.2.2 User Model's Cognition Profile:

Cognition profile includes three main structures: Feedback Mechanism, Adaptive Skills and Intelligence Skills.

Feedback mechanism which is explained in details in following sections is connected with User Model by saving the records of feedbacks that have been transmitted to user up to that time. Due to those feedback results and log analysis, user feedback preference is gathered.

The statistics of adaptive methods – adaptive presentation, adaptive navigation support and adaptive collaboration support – and intelligence methods – curriculum sequencing, intelligent analysis of student solution, interactive problem solving support and example-based problem solving support - that have been applied to user is also held in cognition profile of user modeling.

Pretest score, posttest score, tutorial score, learning gain and other results of learner are kept in evaluation structure of user modeling. Finally the result of questionnaires that have been applied to users in order to get some feedback for our thesis is also a part of user modeling.

3.3 Agent Modeling

Agent model is the subset of User Model. This model only applied to Supportive agents among the agents in framework. Since supportive agents are counted as learners who do not have any adaptive and intelligence skills, most of the design is inherited from User Model except Cognitive profile. Agent Profile is composed of similar two data structures: Agent Profile and Agent Knowledge. All the necessary data in Agent Profile is filled out automatically when system admin creates a new Supportive agent. Initial knowledge level is

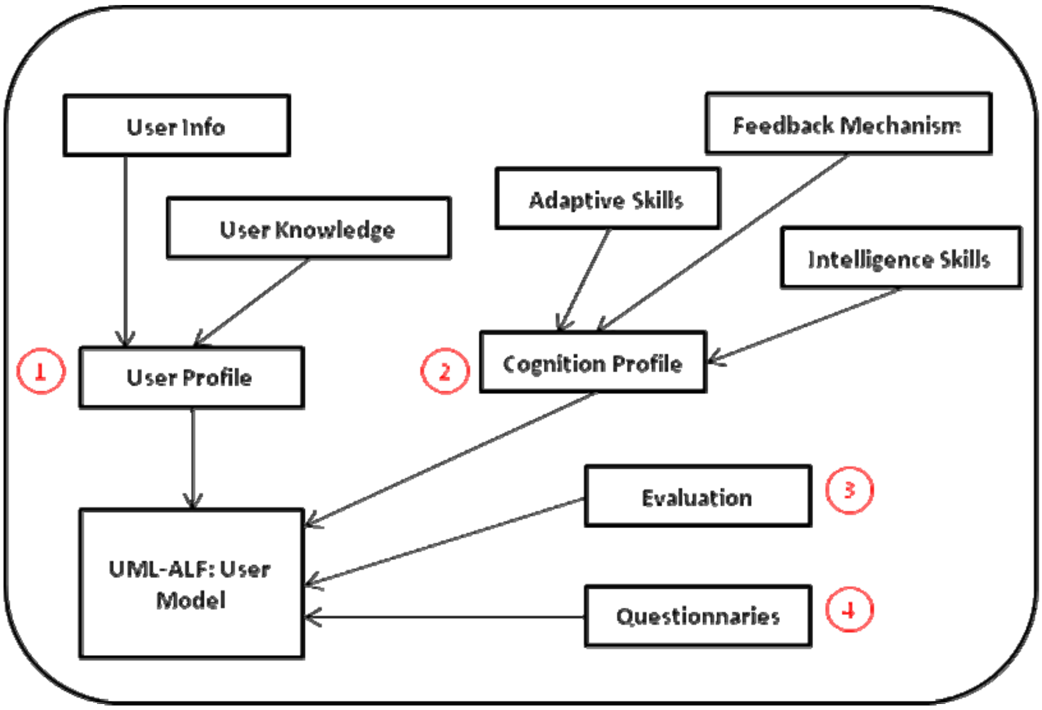

Figure 3.1 User Modeling

also defined during the creation process. There are two kinds of supportive agents according to their goals. Their goal might be helping to the learner or hindering. There is no learning path for supportive agents. They can learn by supporting or learn by collaborating. Learn by supporting is the case when they give some hint to the learner. Learn by collaborating is the situation that they learn from user's decision without helping them. In either way supportive agent should join to the practice in order to improve its knowledge.

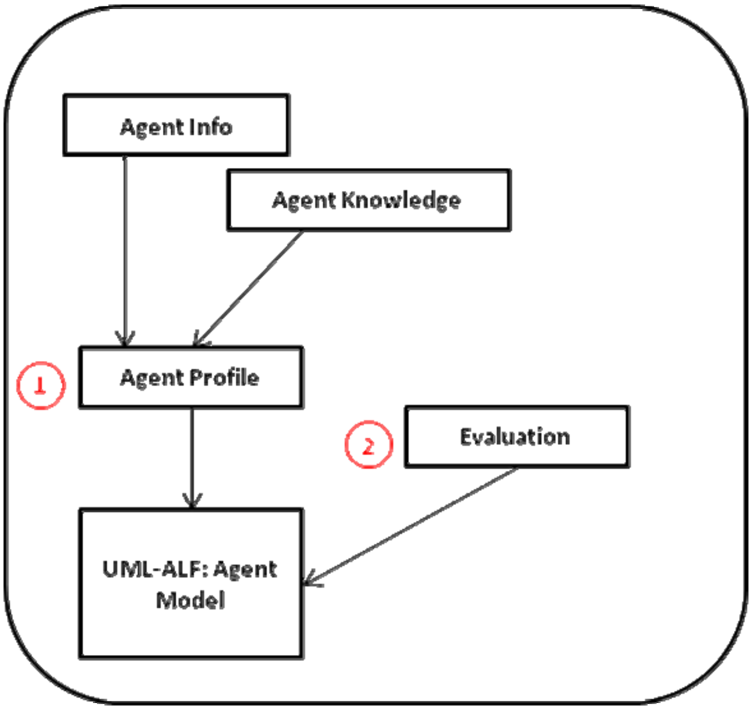

Figure 3.2 Agent Modeling

3.4 Framework

The framework can be abstracted as interface layer, coordination layer and data layer. Interface layer is constituted with User profile, Feedback Mechanism, Supportive agents and User Interface. Feedback mechanism generates effective feedbacks evaluating user's actions in practice mode. Supportive agents help learner to achieve the exercises successfully in practice mode. User Interface is the content viewer in education mode. Content viewer is the place where learning materials are presented to the learner. It becomes the exercise viewer in practice mode. Feedback mechanism and Supportive agents are fed by

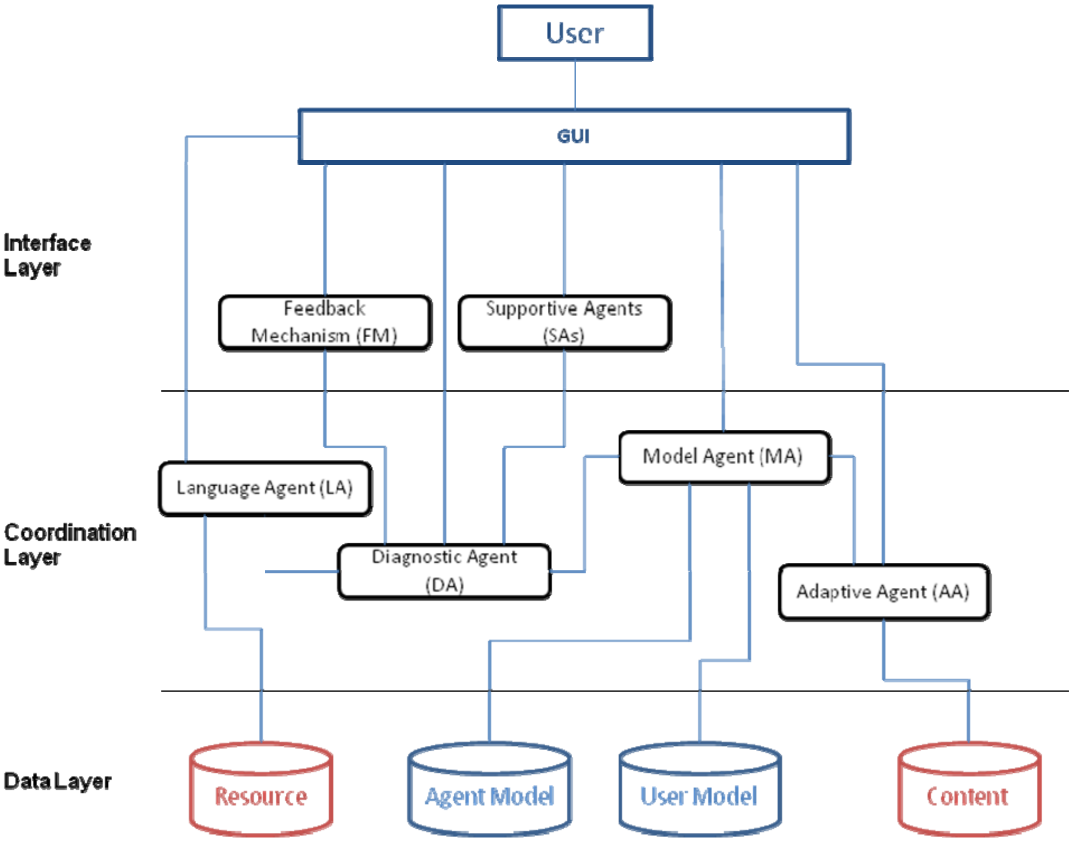

Figure 3.3 Framework Layer

Diagnostic agent which is in coordination layer. User inputs in exercise viewer are transferred through the Language Agent to the Resource. Language agent also provides the communication between Resource in data layer and Diagnostic Agent. Adaptive agent is responsible for displaying learning material according to the adaptive skills of learner. It gets the information of learner with the help of Model Agent which has an access to User Model in data layer. Adaptive agent also gets course data from data layer. Model agent's other duty is to acquire supportive agent data from Agent Model. It sends this information to Diagnostic Agent which maintains the learning of supportive agent.

3.5 AICC

Contents are presented in this study using AICC standards. The Aviation Industry

Computer-Based Training Committee (AICC) is an international association of technology-based training professionals. AICC specifications are intended to be general purpose. Thus, learning content manufacturers may produce their product for different branches of industry at a lower cost. This is the main reason of AICC being so popular today regardless of being in aviation or non-aviation industry. It is first created in 1988 by Aircraft companies in order to standardize the learning materials. In 1993, AICC produced Computer Managed Instruction which is widely accepted as the first runtime interoperability specification. It was first used for local environment. With the arrival of the web, and widely spreading use of browsers, in 1998 AICC added a browser-based format for transferring information in the files. For web compatibility, AICC includes two main communication protocols. HTTP AICC Communication Protocol (HACP) and Application Programming Interface (API). In this research, HACP is chosen. In HACP the files of learning material are packaged as a web page. Then, this package is posted to the server. The learning material can get information from the server or send results to the server. Server responds the request of learning material with plaint/text message as an AICC file. AICC files have a data structure in which information of learning material is saved and stored. This data structure includes Core (session id, score, date, exist status, lesson location), Core Lesson, Student Preferences, Objective, Interactions and Paths.

3.6 Adaptive and Intelligent Techniques

In learning material, Adaptive Presentation technique is used in order to personalize the content. According to Brusilovsky, P. (2001), adaptive presentation is grouped in three categories: the adaptation of multimedia, the adaptation of modality and adaptive text presentation. The adaptation of multimedia deals with manipulating the quality and size of the multimedia content like video or image. The adaptation of modality is to select the media to be used. This study focuses on the adaptive text presentation category of adaptive presentation. The categorization of Brusilovsky, P. continues with two types of adaptive text presentation: Natural language adaptation and canned text adaptation. Natural language adaptation which is based on natural language understanding is beyond of this study. The forms of canned text adaptation are used in UML-ALF. According to the characteristics of users, the presentation of content is changed by adding or removing pages, hiding some parts in different ways in this work. Due to the result of pre-test and values of user knowledge structure, system creates a learning path which includes learning materials specifically

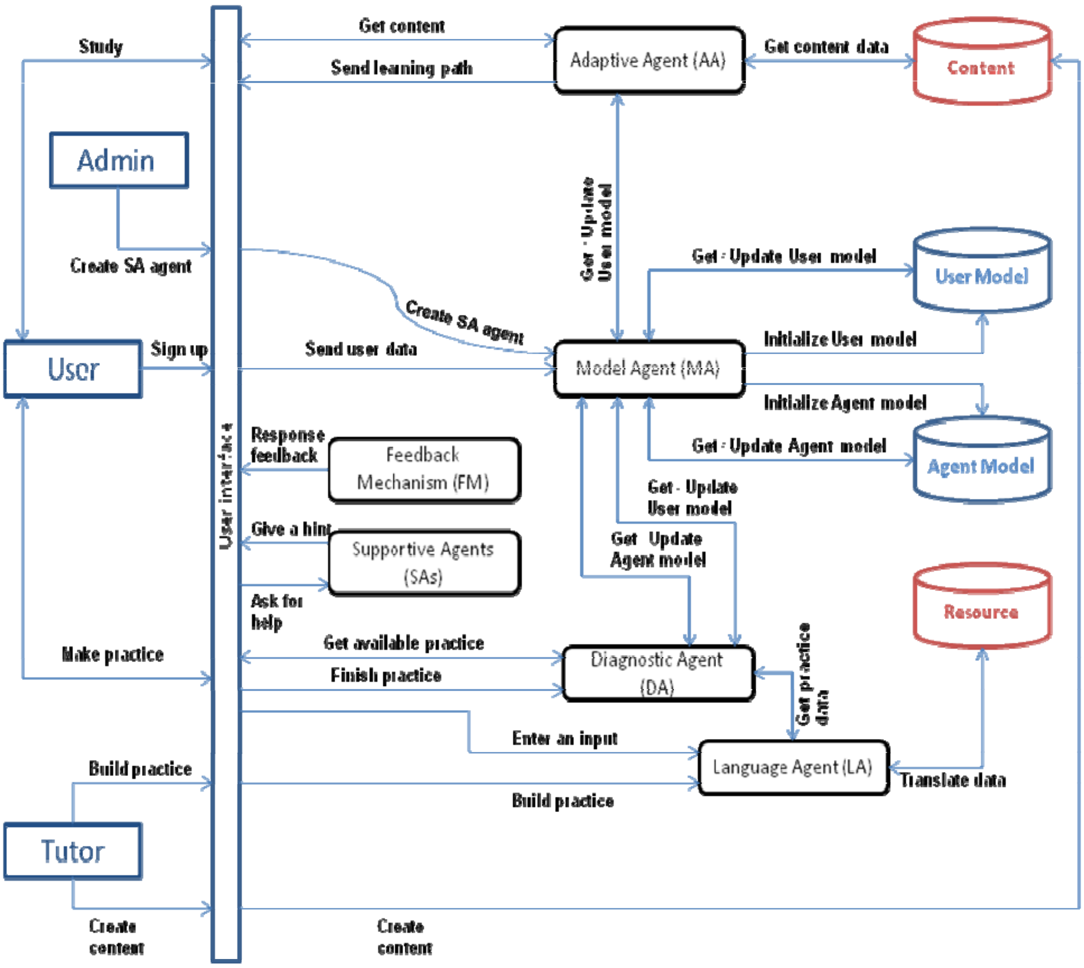

Figure 3.4 Workflow of the framework

selected for that user using curriculum sequencing skill. According to the knowledge base of user, unnecessary topics are hidden from the learner on his learning path. With the help of adaptive navigation support, system guides the user to follow his learning path easily using direct guidance, link sorting and link annotation techniques.

While the user is being examined of his/her knowledge or making practice, system responses a detailed feedback which is obtained from feedback mechanism to student using intelligent analysis of student solutions skill. Also interactive problem solving support helps user during the solution period of the problem following the same protocol with the help of Supportive Agents. Supportive agents are agents which act like student teammates. They not only respond to user's answers but also intervene during the solution process. Another support while solving a problem is misleading hint support which are common mistakes of other learner’s. Learners are informed that some agents might give misleading hints. Also, tutor agent provides the correct hint using scaffolding approach after supportive agent’s suggestion.

3.7 Intelligent Agents

Six different intelligent agents are used in this framework: Diagnostic Agent (DA), Language Agent (LA), Feedback Mechanism (FM), Supportive Agents (SAs), Model Agent (MA) and Adaptive Agent (AA). All of these agents are rational agents which does the right thing. Here, doing the right thing is taking the right action which makes the agents most successful. In order to measure the success, an evaluation is needed. Thus, the rationality of an agent depends on four things (Russell & Norvig, 2003):

- Evaluation of success
- Initial knowledge of an agent
- Actions that an agent perform
- Perception of an agent

The task environment of an intelligent agent system is determined by combination of four terms – Performance, Environment, Actuators and Sensors which is called PEAS shortly (Russell & Norvig, 2003).

The task environment of this study is **partially-observable (inaccessible)**. Since each agent’s sensors have no access to complete state of environment. However, whole agent system is fully-observable (accessible). It is **stochastic (non-deterministic**), because one can never predict the behavior of users and tutors. Another property of the environment is that it is **sequential (non-episodic)**. Each step affects all future steps. Moreover it is a **dynamic** environment. As the time flows, things change. On the other hand, time does not flow continuously. It proceeds step by step. So, it is a **discrete** environment. The final property of the environment is that it is a **cooperative multi-agent** environment. Agents are not competitive, even they help each other.

3.7.1 Diagnostic Agent (DA):

Diagnostic agent is a model based reflex agent. It is the center of the intelligent system. As the user makes progress through the interface, it gets translated rule package from Language Agent (LA). Then it diagnoses the rule package by communicating resource. It also can determine that user's answer is an alternative solution to the question even it is not stated in the resource. It can automatically corrects typos that user makes. Then it generates reactions. These reactions may be a general hint, a crucial hint, a complete answer or a misleading hint. Then sends those set of reactions to supportive agents (SAs). After user decides to finish the practice, solution is translated to rule packages again by Language Agent (LA). Then DA evaluates the rule packages and sends the results to the Feedback Mechanism (FM).

3.7.2 Language Agent (LA):

Language agent is a simple reflex agent. The raw data from user interface of User or Tutor profiles come to LA. It takes those raw data and translates them into meaningful –for agents- rule packages. When user makes a progress or finishes the practice and tutor creates a practice, the data is needed to be translated by LA.

3.7.3 Feedback Mechanism (FM):

Feedback mechanism is the structure where feedback messages are generated. Feedback can be classified as positive and negative feedbacks. Negative feedbacks are delivered in response to mistakes to help correcting them and positive feedbacks are delivered in response to correct input. Those feedbacks might be verbal or non-verbal feedbacks. Verbal feedback can be written or spoken whereas non-verbal feedbacks are generally mimics, gestures, sounds or pictures.

There are 3 kinds of feedbacks: Failure feedback, Success feedback and warning feedback. Failure feedback is not a simple feedback message. For a detailed feedback message that includes some hint to solve the problem, neural networks are used. Especially success feedback messages consist of both verbal and non-verbal feedbacks.

Warning feedbacks are neither failure nor success messages. In fact, they are not related with the result of solution. Warning feedbacks are generated when system could not understand or fulfill the request of the user. Also they might be system error, incompatibilities, improper usage, etc.

Feedback Mechanism is also a simple reflex agent. After user finishes the practice, consecutively LA translates the raw data to rule packages. Then DA diagnoses the rules and sends results to FM. If the result is correct, it returns a positive feedback. Otherwise, it

Table 3.1 Description of the task environment

Agent Type	Performance Measure	Environment	Actuators	Sensors
Diagnostic Agent (DA)	successful reactions, misleading reactions, auto completion, alternative solutions	Resource, FM, LA, SAs, User	check data, generate reaction, send feedback data	rule package (LA), finish practice (User)
Language Agent (LA)	translation	User, Tutor, DA, Resource	translate	raw input data (User, Tutor)
Feedback Mechanism (FM)	positive feedback, negative feedback	DA, User	response feedback	feedback data (DA)
Supportive Agents (SAs)	misleading response, no response, helpful response	User, DA	give hint	ask for help (User), generate reaction (DA)
Model Agent (MA)	initializing user model, initializing agent model	User, Admin, User Model, Agent Model, DA, AA	create user profile, create SA, response User Model, response Agent Model	sign up to the system (User), create SA (Admin), get User Model (AA), get Agent Model (DA)
Adaptive Agent (AA)	adapting learning material	User, Course, MA	send course content, get course raw data, get user model	get course content (User), send course raw data(Course), send user model (AA)

examines the data and creates an intelligent negative feedback, which tells user clearly where he/she made a mistake. Feedback mechanism has three outputs: sound, gesture and text. Except the text it produces, it also makes a sound effect and responses a proper gesture.

3.7.4 Supportive Agents (SAs)

Supportive agents are agents which act like student teammates. They not only respond to user's answers but also intervene during the solution process. Basically they have three main actions. They can help by giving useful suggestions. Those suggestions may be crucial or might not help at all. Secondly, they may not interfere or they have no opinion about that situation. Lastly, they can hinder and convince user with their opinions. All of these are possible teammate behaviors in collaborative work. User may force the irrelevant SA to comment on subject. If it has knowledge about subject, may give correct information or if it has no idea, it make up an answer which is not correct. User also may force other SAs - helping SA or hindering SA- to stop or to keep on giving advices. The actions of simulated collaborative students – SAs - are modeled using artificial intelligence methods such as Neural Networks and Hidden Markov Model.

Learning from mistakes is one of the learning methodologies. Trying to mislead learner with a common mistake is also used in (Mengelle et al., 1998). The benefit of learning by making mistakes is also discussed in (Dweck, 2007; Izzo & Mazzone, 2008; Chialvo & Bak, 1999). As a similar approach, “trial and error”, or “trial by error” or “try an error”, is a general method of problem solving, fixing things. In the field of computer science, the method is called “generate and test”. In elementary algebra, when solving equations, it is "guess and check". Therefore, it is decided to implement suggesting a common mistake by supportive agents as an alternative option.

SAs are goal-based agents. They may have two goals: misleading the user or helping user. When user asks for help, related SA gets the reactions from DA and decides its action among the reactions. It automatically gives the hint to the user. However, when user makes a progress, each SA may have an opinion about that move, but cannot tell directly what it thinks. Instead, it blinks which means it has something to tell. User may allow telling its opinion or not.

3.7.5 Model Agent (MA)

Model agent (MA) is a model based reflex agent. When user signs in to the system at first time, user encounters pre-tests and questionnaires. Due to the results of these tests, Model Agent initializes the model of the user. It sets up the knowledge level of the user, determines his/her preferences and creates his/her learning path. When Adaptive Agent displays the learning material to the user, it asks MA for the necessary information about the user model and MA sends the information back. Another duty of MA is to initialize the agent model for the supportive agent after system admin determines the information of newly created SA. While supporting user in practice area, Diagnostic Agent asks MA for related SA's agent model and MA sends the information back.

3.7.6 Adaptive Agent (AA)

Adaptive Agent (AA) is a model based reflex agent. The main duty of AA is to display the learning material after adapting it according to the user's user model. It gets the necessary information of user from MA.

CHAPTER 4

METHOD

4.1 Implementation & A Case Study: UML

4.1.1 Unified Modeling Language (UML) Domain

As software projects become larger, programming complexity will increase. Moreover, the number of programmers in the project will increase as well. Therefore, a necessity for standard modeling and design language appears. Unified Modeling Language (UML) is a collection of methods which explain and define how to model work flows, rather than being a programming or software development language. It is standardized and created by Object Management Group.

The UML model and the set of diagrams of a system should be differentiated. A diagram is one of the parts of system model which is visually represented. There are totally 14 diagrams in UML 2.2. Those diagrams are grouped under two main categories: Structure (Static) Diagram and Behavior (Dynamic) Diagram. Structure Diagrams and their features are shortly described as:

Class diagram: is one of the most used UML diagram. It aims to represent classes that constitute the basis of Object Oriented Analysis, Design and Programming.

Component diagram: It shows the components that constitutes the software system and dependencies among other components.

Composite structure diagram: describes the internal structure of a class and the collaborations that this structure makes possible.

Deployment diagram: serves to model the hardware used in system implementations, and the execution environments and artifacts deployed on the hardware.

Object diagram: shows a complete or partial view of the structure of a modeled system at a specific time.

Package diagram: depicts how a system is split up into logical groupings by showing the dependencies among these groupings.

Profile diagram: operates at the meta model level to show stereotypes as classes with the <<stereotype>> stereotype, and profiles as packages with the <<profile>> stereotype. The extension relation (solid line with closed, filled arrowhead) indicates what meta model element a given stereotype is extending.

Behavior diagrams have following diagrams:

Activity diagram: represents the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

State machine diagram: standardized notation to describe many systems, from computer programs to business processes.

Use case diagram: shows the functionality provided by a system in terms of actors, their goals represented as use cases, and any dependencies among those use cases.

Since behavior diagrams illustrate the behavior of a system, they are used extensively to describe the functionality of software systems.

Interaction diagrams: a subset of behavior diagrams, emphasize the flow of control and data among the things in the system being modeled:

- **Communication diagram:** shows the interactions between objects or parts in terms of sequenced messages. They represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.
- **Interaction overview diagram:** is a type of activity diagram in which the nodes represent interaction diagrams.
- **Sequence diagram:** shows how objects communicate with each other in terms of a sequence of messages. Also indicates the lifespan of objects relative to those messages.
- **Timing diagrams:** Is a specific type of interaction diagram, where the focus is on timing constraints.

4.1.2 Training Process of UML-ALF

The training process of UML-ALF starts with registration of user to the system. User information, additional preferences and settings for experimental procedure is defined during the registration.

System is proposed of two main parts.

Book image represents the Study Part of the system. This part is managed and applied with the help of the adaptive techniques which are introduced in previous chapter. User can take online courses and solve post-tests which are adapted according to their characteristics and knowledge level.

Pencil image represents the Practice Part of the system. This part is managed and applied with the help of the intelligent techniques. Three UML diagrams – Class Diagram, Use – Case Diagram and Activity Diagram- are chosen as practice models. Users can improve their own skills with the help of this interactive practice screen, which can suggest new ideas, evaluate and track user moves in very close manner.

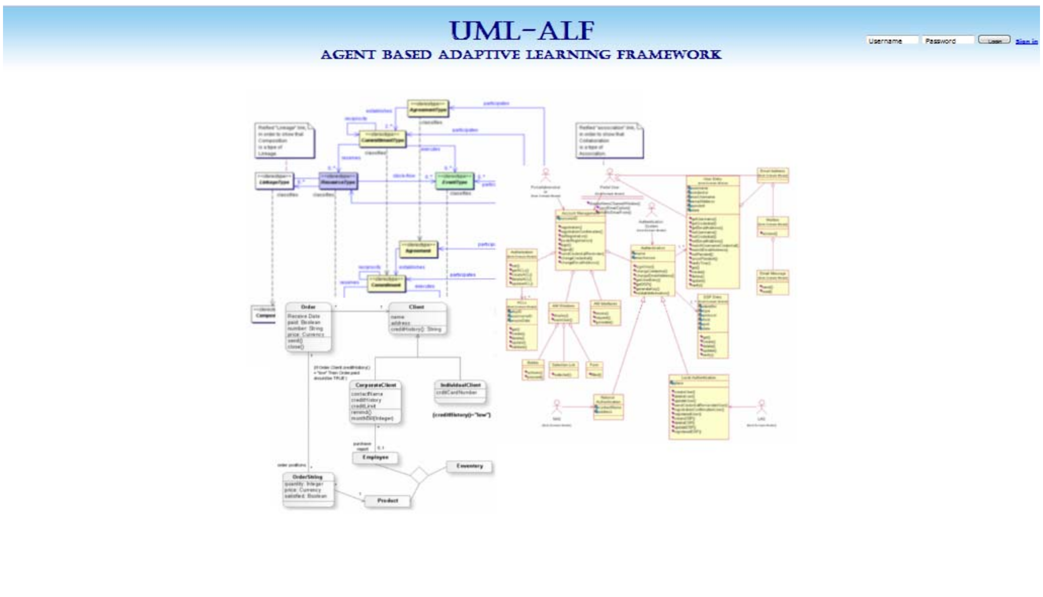

Figure 4.1 UML-ALF Home Page

After signing to the system, at first time user is introduced with pre-test (Figure 4.4). According to the result of pre-test, a proper learning path (Figure 4.3) is assigned to the user

Figure 4.2 UML-ALF Registration page

by the adaptive agent. A training package includes e-learning material, post-test and intelligent practice environment. User knowledge model determines the courses that user may attend. Green color shows the training packages which was already completed. Orange color shows the unlocked materials whereas Red color implies the locked materials. (Figure 4.5)

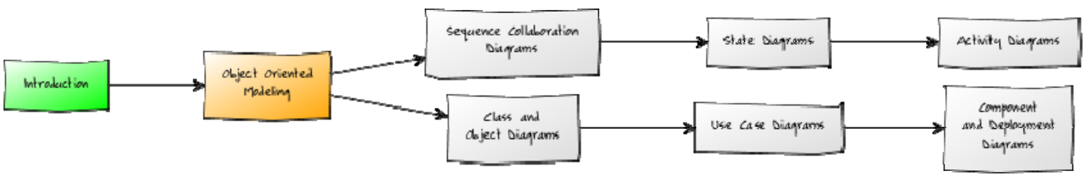

Figure 4.3 UML-ALF Learning path

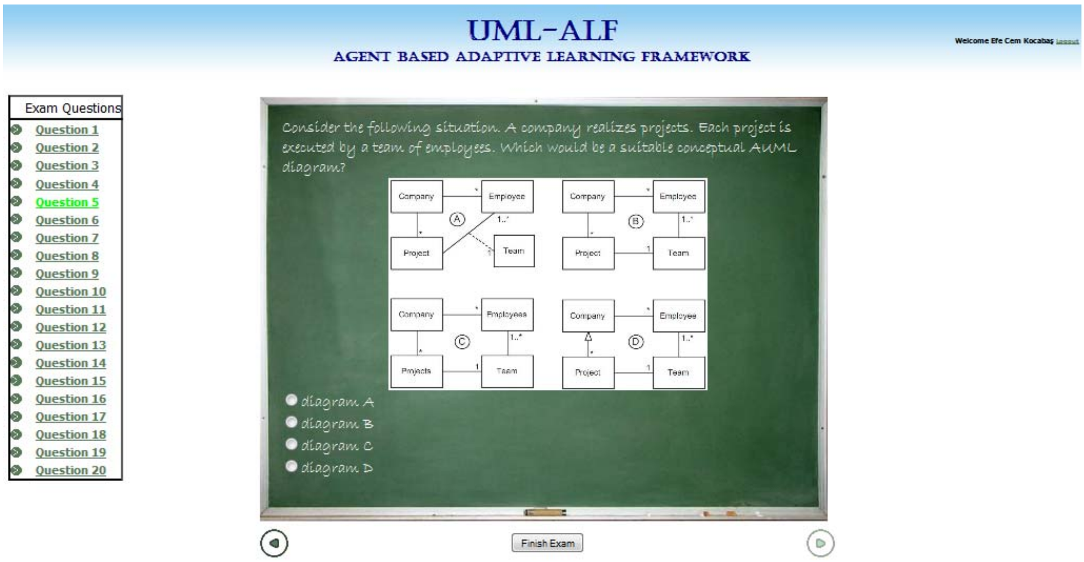

Figure 4.4 UML-ALF pre-test

UML-ALF has eight chapters while training the UML domain: Introduction, Object Oriented Modeling, Class and Object Diagrams, Use Case Diagrams, Component and Deployment Diagrams, Sequence and Collaboration Diagrams, State Diagrams and Activity Diagrams. Each chapter is formed of an e-learning course which is designed by following AICC standards and a post-test which evaluates the knowledge gain of user from online course. Also three chapters - Class Diagram, Use – Case Diagram and Activity Diagram – has an additional practice link that redirects to Practice Part of the system. E-learning materials are also prepared with adaptive techniques. If user has mastered some parts of the chapter but not the whole, then when he/she experiences the same course second time, only the parts which were not fully learned before are introduced to the user.

There is no limitation for taking courses over and over again. If the user has already completed the course before, system warns the user about whether taking the course with full parts (without adaption) or not. Detailed information about the process of taking e-learning materials is logged and related charts can be produced in admin profile of the system.

Practice part of the system uses intelligent techniques of artificial intelligence. Users may improve their knowledge level by practicing the subjects learned in Study Part. In practice screen user is asked to generate a UML diagram using the information given on top of screen (Figure 4.7 item-3). Learners often demand feedback, tutorial and guidance in learning environments. Generally these requirements are being provided with the help of human–computer interaction in computer-based learning environments. (Kızılkaya & Aşkar, 2008). There are three supportive agents that help user during the practice period (Figure 4.7 item-10). Two of these agents always give the correct suggestions as long as they know the

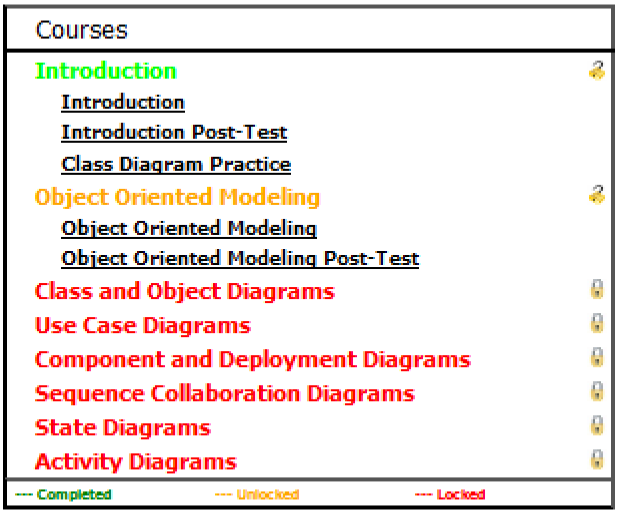

Figure 4.5 UML-ALF Course Tree

subject. However one of them suggests a common mistake which has been made in that practice before. Last agent is a tutor who stands at the right-top of the screen (Figure 4.7 item-9). When an agent makes a comment about a move that user made or suggests a new idea, tutor evaluates this information and gives the most correct and definite result to the learner. Firstly, user decides which UML diagram is used for that practice, then from toolbar (Figure 4.7 item-7) at the left side of the screen, he/she chooses the necessary elements. Those elements are placed to the practice area by drag & dropping. If it is an element that

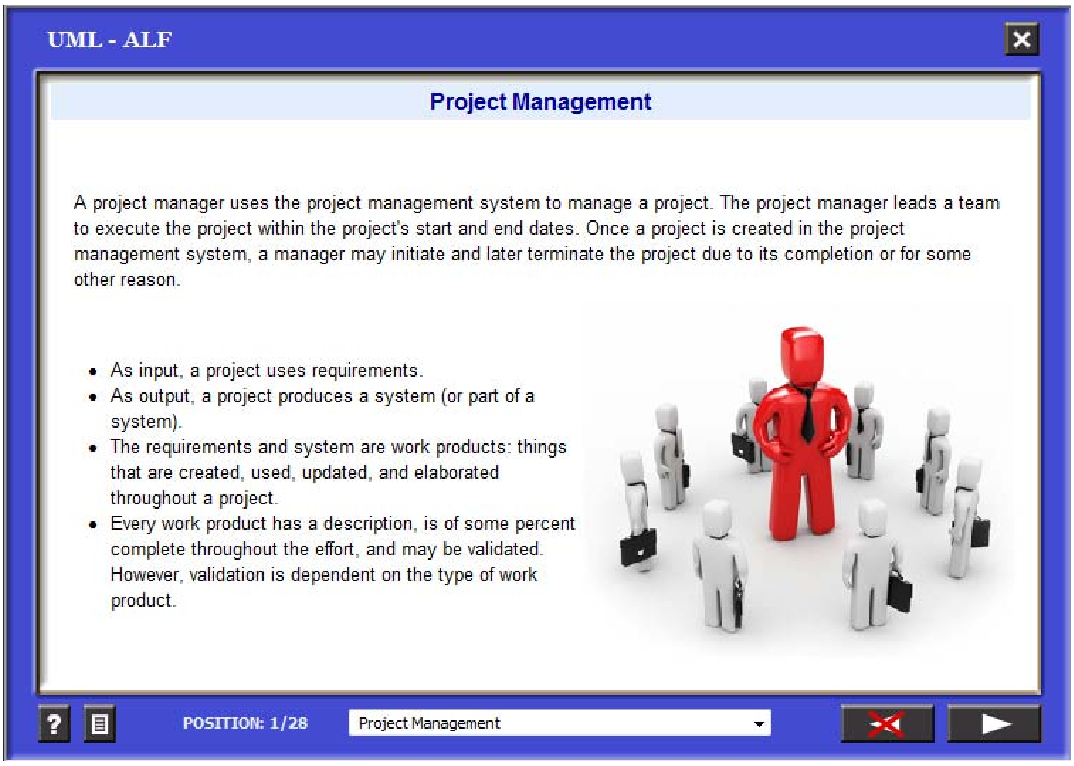

Figure 4.6 First page of Object Oriented Modeling Course

can be named, user should change the default name of the element to the correct name according to the scenario of the practice. With the relation tools, like directional, aggregation, composition, etc. the relation between the elements can be provided. Also, if necessary, cardinality of the relation may be defined.

After each move of the learner, supportive agents may comment about that move. They indicate their desire of making comment by blinking with gray colors. After they show their will, user may click on agent's photo to learn about their idea. They express their opinion with a speech balloon. Finally, tutor agent makes a decision about whether the expression of supportive agent is true or not. He shows his decision by nodding his head and with the color of his picture. If he nods his head left to right and the color is red then he means the expression is not correct. If he nods his head up and down and he color is green then he means the expression is correct.

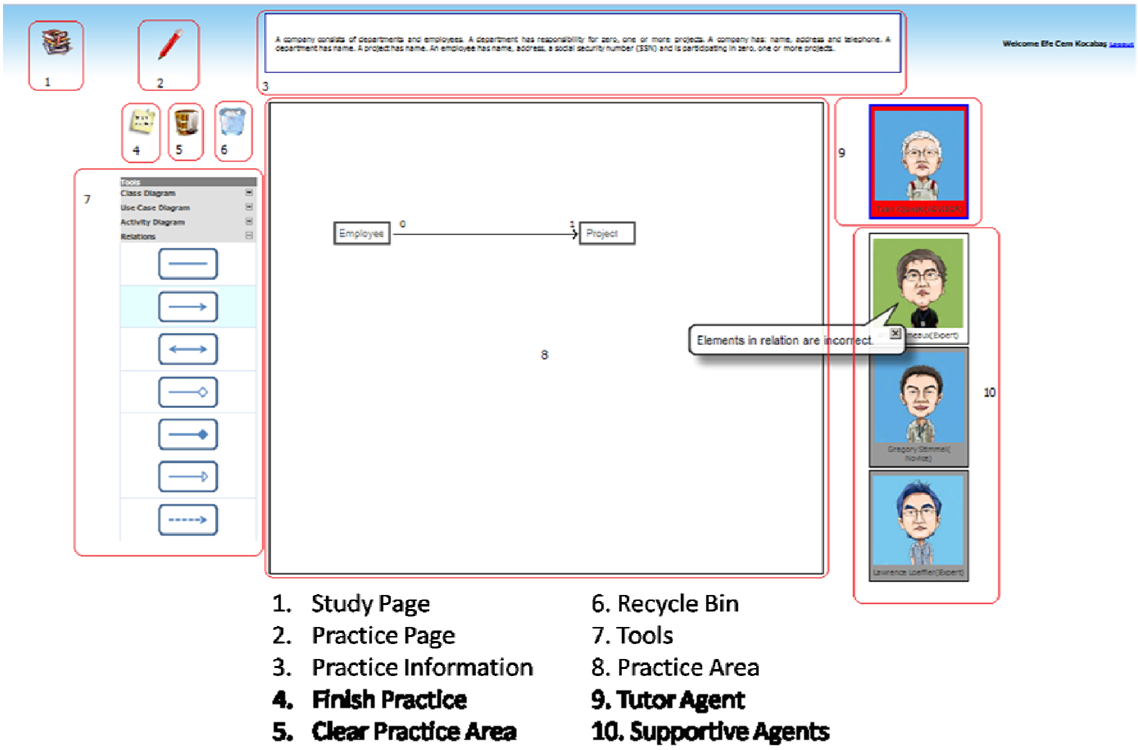

Figure 4.7 UML-ALF Practice Screen

Dragging an element clears its relations with the other elements and dragging an element to the recycle bin disposes it. The icon that is numbered with 5 in figure 4.7 clears whole practice area. After learner decides that he/she is done with the practice, he/she presses the 'Finish Practice' button to learn the result of the practice. The score of user and the correct solution of the practice is shown on a popup window. Supportive agents have their own knowledge model. Therefore, as they made new comments and attend new practices, they improve their knowledge level. There are totally 2 tutor agent and 10 supportive agents. Since they are goal-based agents, they are classified due to their goals. Three of them aim to frustrate the learner and rest of them are helpful agents.

Every move of learner is being tracked and logged by the system in order to evaluate precisely. When learner makes same mistakes again and again, system thinks that there would be a problem and asks a relevant question to the learner. If learner answers this question correctly, he/she goes on practice; otherwise a summary of the relevant topic is shown to the learner. Then, another question is asked. If learner still cannot find the correct answer, finally he/she is redirected to the course of the subject.

4.1.3 Building Resources

The resource is in charge of the diagnosis of student knowledge and informs the actors about what they can do next. So, clearly, the global quality of the ITS also depends on the expertise of the resources. Thus, it is decided to build a resource structure.

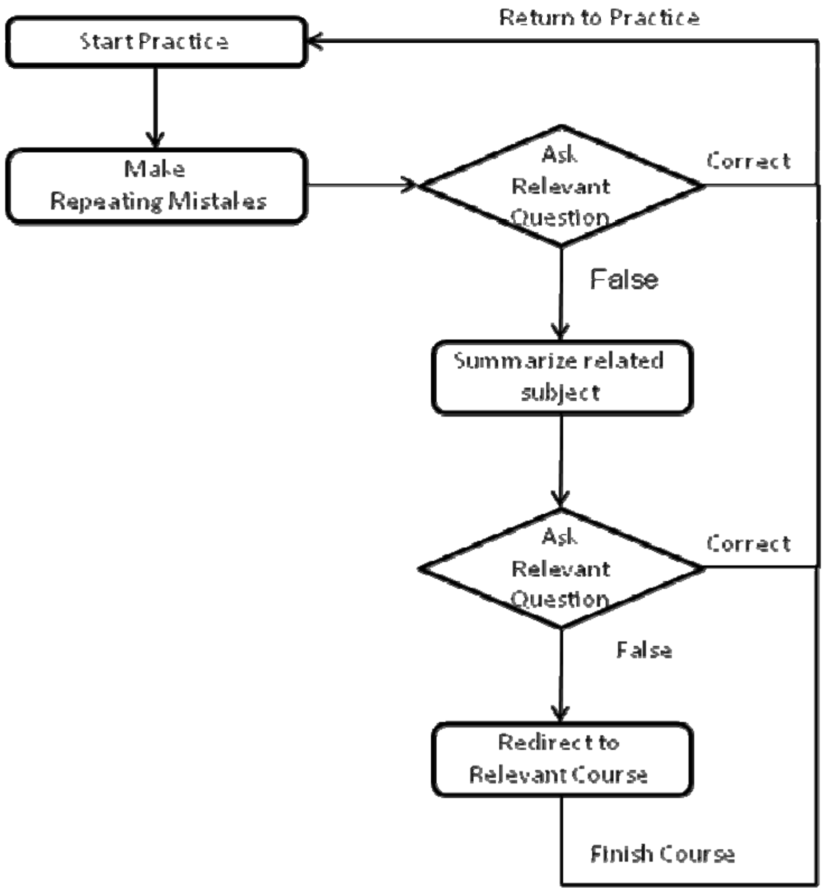

Figure 4.8 UML-ALF User Tracing Flow Chart

The resource structure consists of three main parts:

- Subject Information: an identification of resource, UML diagram type, the list of related subjects and question text.
- Answer Rules: it is the whole solution of the practice. It is initially determined by the designer of the practice.
- Practice Hints: a set of hints regarding related subjects.

Tutor is the designer of a practice. The resource can be inserted to the system either through user interface or via an XML file. An example of a resource structure data can be found in Table-4.1 and its output is stated in Figure 4.9.

Table 4.1 Resource structure

<table>
<tr><td>Resource ID</td><td>1</td></tr>
<tr><td>UML Diagram Type</td><td>Class Diagram</td></tr>
<tr><td>Related Subjects</td><td><class_diagram>,<directional>,<aggregation>,<cardinality></td></tr>
<tr><td>Question Text</td><td>A company consists of departments and employees. A department has responsibility for zero, one or more projects. A company has: name, address and telephone. A department has name. A project has name. An employee has name, address, a social security number (SSN) and is participating in zero, one or more projects.</td></tr>
<tr><td>Answer Rules</td><td>[Employee|name;address;SSN]
[Company|name;address;phone]
[Project|name]
[Department|name]
[Company] + 1 - > * [Employee]
[Company] + 1 - > * [Department]
[Department] – 0... * > [Project]
[Project] – 0... * > [Employee]</td></tr>
<tr><td>Hints</td><td>Actors are not an element of Class Diagrams.
The relationship between those subjects is not ‘Composition’.
Watch out the cardinalities!</td></tr>
</table>

4.2 Definition of Propositions

1. Supportive agents provide contribution to the learning achievement of learners in distance education.
2. Learners are interacted with supportive agents as it is purposed.
   2.1 The methodology that agents give misleading hints which are common mistakes of other learners had positive effects on learning the subject.
   2.2 The methodology which forces learners to renew their knowledge as they make mistakes is effective on their learning achievement.
3. Intelligent agents and computer based tutors can give effective feedback to the learners which is close to the performance of human tutors.

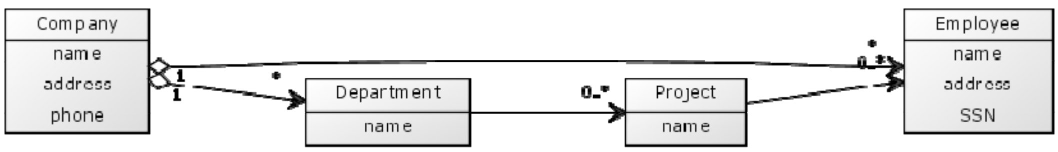

Figure 4.9 Output of the resource structure stated in Table 4.1 (Harris, 2009)

4.3 Participants of the study

In order to assess that UML-ALF has accomplished its objectives, it is decided to follow an experimental procedure. Three different features of this framework were considered during the experiment. Those features are adaptive hypermedia, intelligent tutoring system and advanced feedback mechanism.

Currently, the proposed UML-ALF is available on the web to provide personalized UML learning. In order to verify the quality of UML-ALF, people were invited to test this system. Purposive sampling method (Patton, 2002) was preferred to choose the suitable sample for our experiment. The main idea is to gather two groups one of which is experts of the subject and the other one is willing to learn the subject but has no background.

Before testing the system, participants joined a survey [Appendix B] that determines in which category they should be included. According to the results of survey people are gathered in two groups. These groups are named as Experienced and Inexperienced. People in experienced group has advanced knowledge about UML, they took a UML related course in their education, use the benefits of UML diagrams in their current work. On the contrary, participants in inexperienced group are lack of UML knowledge or little if any, however they are interested in subject. In each category, participants were divided into two sample groups. While first group had no benefit of UML-ALF's features, second group has taken advantage of those features. A total of 32 participants were analyzed and reported within the scope of this study. Demographics of the participants were not taken into consideration, since the experiment is only interested in learning background of the participants.

4.4 Data Collection Procedure

When carrying out a study where data acquisition is needed from the human participants, an approval from the Applied Ethics Research Center is obligatory at METU. Therefore, within the scope of this study, an ethical consent is prepared and an approval from the ethics committee is obtained before data gathering process. (Appendix E). Each participant in this research has signed the ethical consent before participation.

Before performing the test to whole sample, a pilot study is conducted. The pilot study is carried out with 19 individuals. Participants joined the survey [Appendix B] that determines in which category they should be included. According to this survey; Experienced Group has 7, inexperienced group has 12 participants. Those participants are gathered from the calls that are announced in newsgroups of Computer Engineering Departments of the METU and IYTE and other communication newsgroups and forums. Participants made use of all system features. All courses and one practice were available during pilot test. The aim of the pilot study was to simulate the data gathering process and to foresee any problems that might be faced during data gathering.

The results of the pilot test was evaluated in two parts. Table 5.1 shows the results of adaptive tests. The number of participants answered the test, the number of minimum and maximum correct answers, median and standard deviation for each chapter are stated in Table 5.1. According to the this table number of participants were decreased as time goes by. It is concluded that participants has lost their interest. It is decided that one major cause of this might be the high threshold that is defined for moving from one unit to the next. The

Table 4.2 Adaptive Results of Pilot Test

Test	Number of Participants	min	max	M	SD
Introduction pre-test	19	1	8	4,16	2,03
Introduction post-test	17	2	10	6,82	2,04
Object Oriented Modeling pre-test	10	1	8	5,6	2,33
Object Oriented Modeling post-test	9	3	9	7,56	1,77
Class and Object Diagrams post-test	8	3	7	5,38	1,32
Class and Object Diagrams pre-test	8	4	9	7,13	1,62
Use Case Diagrams pre-test	6	2	8	5,65	1,87
Use Case Diagrams post-test	4	4	9	6,9	1,88
Component and Deployment Diagram pre-test	5	4	8	6	1,41
Component and Deployment Diagram pre-test	4	4	10	7,5	2,18
Sequence and Collaboration Diagrams pre-test	2	3	4	3,5	0,5
Sequence and Collaboration Diagrams pre-test	2	5	10	7,5	2,5
State Diagrams pre-test	3	2	6	4	1,63
State Diagrams post-test	1	6	8	7,34	0,94
Activity Diagrams pre-test	1	4	4	4	0
Activity Diagrams post-test	1	6	6	6	0

score threshold was lowered from 85% success to approximately 70% success for each unit.

The results of practice part of the system showed that related parameters for evaluating the success of intelligent agent techniques such as logging learner's moves and agent's responses did not constitute enough data for the researcher. Thus, it is decided to observe participant's moves in Human Computer Interaction laboratory and one-to-one experiments.

4.5 Data Analysis Methodology

As previously stated, most e-learning systems focus on intelligent tutoring methodologies, but ignore basic learning theories. However, including these theories to learning environments is highly crucial, since they provide clearer images of the students' learning circumstances. Therefore, learning systems could adapt according to learner's characteristics. Keller & Sherman (1974) proposed a study model called Personalized System of Instruction (PSI) for school education which allows students to follow a lecture at an individual speed. System divides the whole content into basic learning elements and each of these units has a content and a test. A score threshold is defined for proceeding to the next unit and a student must get enough score to move from one unit to the next. In our study, *Keller's Personalized System* was adopted.

In most evaluation systems, student's score is set according to the major assessment rules. Another factor regarding quality of learning is learning achievement. It is stated that

Table 4.3 Content of the Implemented Course UML

UML			
Chap.	**Subject**	**Chap.**	**Subject**
1	Introduction	5	Component and Deployment Diagram
2	Object Oriented Modeling	6	Sequence and Collaboration Diagrams
3	Class and Object Diagrams	7	State Diagrams
4	Use Case Diagrams	8	Activity Diagrams

Table 4.4 Snapshot of the Learning Nodes of Chapter 3

Chapter 3 – Class and Object Diagrams	
Topic ID	**Contents**
1	Classes and Objects
2	Associations and Links
…	
7	Packages and Subsystems

most effective learning performance is determined by using assessments and test (Scott, Beichner, Titus & Martin, 2000; Shih, Lin & Change, 2003). The assessment or test generally produce two consequences. The first one lets the learning system or tutors analyze the progress in achieving the learning objective, and the second one gives effective feedback to the learners. However, the conventional test theory does not respond easily to individual performance. In this study, the modern test theory is applied in order to obtain each student's learning achievement.

The practice part of the UML-ALF has been evaluated in two ways. 5 participants has been experienced UML-ALF in Human Computer Interaction Laboratory in METU. The purpose of this experiment was to research the nature of interaction between participants and UML-ALF in order to enhance the design and utilization of more usable and humanly acceptable system. Also, the correlation between supportive agents and student's learning achievement was observed.

In practice part of UML-ALF, the relationship between two variables, the usage of supportive agents by the learner and their practice score was observed. The hypothesis is that how much learner made use of supportive agents affects their practice score. Here, there is no need to worry about the direction of causality, since it's not likely that practice score causes supportive agent usage. Supportive agent usage is a number which increases by 1 each time the learner get helped.

CHAPTER 5

RESULTS

The purpose of this study was to present an agent-based adaptive learning framework whose goal is to implement effective tutoring system with the help of AI techniques. In this section, the results for each research questions are introduced.

5.1 Proposition 1

First proposition is defined as supportive agents provide contribution to the learning achievement of learners in distance education. In order to find the contribution of supportive agents, the correlation between two variables, the usage of supportive agents by the learner and their practice score was based on.

The hypothesis is that how much learner made use of supportive agents affects their practice score. Here, there is no need to worry about the direction of causality, since it's not likely that practice score causes supportive agent usage.

Table 5.1 Descriptive Statistics of Practice Results

Variable	Mean	Standard Deviation	Variance	Sum	Min. Value	Max. Value	Range
Score	87,32	10,26	105,27	2707	62	100	38
Ias	15,32	6,02	36,24	475	5	29	24

Table 5.2 Statistics for correlation between # of Agent Suggestion and Practice Score

Participant	# of Agent Suggestion (x)	Practice Score (y)	x*y	x*x	y*y
1	29	100	2900	841	10000
2	19	81	1539	361	6561
3	26	92	2392	676	8464
4	11	76	836	121	5776
5	22	92	2024	484	8464
6	24	100	2400	576	10000
7	19	96	1824	361	9216
8	18	94	1692	324	8836
9	6	90	540	36	8100
10	13	100	1300	169	10000
11	15	88	1320	225	7744
12	7	66	462	49	4356
13	17	80	1360	289	6400
14	6	62	372	36	3844
15	16	80	1280	256	6400
16	16	96	1536	256	9216
17	8	72	576	65	5184
18	12	92	1104	144	8464
19	18	96	1728	324	9216
20	12	82	984	144	6724
21	16	90	1440	256	8100
22	21	98	2058	441	9604
23	18	86	1548	324	7396
24	9	76	684	81	5776
25	5	72	360	25	5184
26	18	90	1620	324	8100
27	14	94	1316	196	8836
28	22	100	2200	484	10000
29	14	96	1344	196	9216
30	7	82	574	49	6724
31	17	88	1496	289	7744
Sum=	475	2707	42809	8402	239645

Participants that made use the benefits of the intelligent techniques of UML-ALF has produced totally 31 practice results in 3 practices (Class Diagrams, Use Case Diagrams and Activity Diagrams). Symbol ‘r’ is used to stand for the correlation. According to the result of Table 5.6, r is calculated as 0.69. The result of r shows that there is a positive correlation between variables practice score and number of agent suggestion.

Once the correlation was computed, the probability that the observed correlation occurred by chance was determined. That is, a significance test is conducted. In this case, the mutually exclusive hypotheses: null hypothesis and alternative hypothesis was tested.
The significance level is used as common significance level of alpha = 0.05. This means that a test was conducted where the probability that the correlation is a chance occurrence is no more than 5 out of 100. The degrees of freedom (df) is simply equal to N-2, in this study it is 29. Finally, it is decided to do two-tailed test, since it is not clear that the relationship between practice score and number of agent suggestions would be positive or negative. With these three parameters - alpha = 0.05, df = 29 and type of test is two-tailed, using the lookup table of critical values for Pearson’s r, the critical value is found as 0.355. This means that if

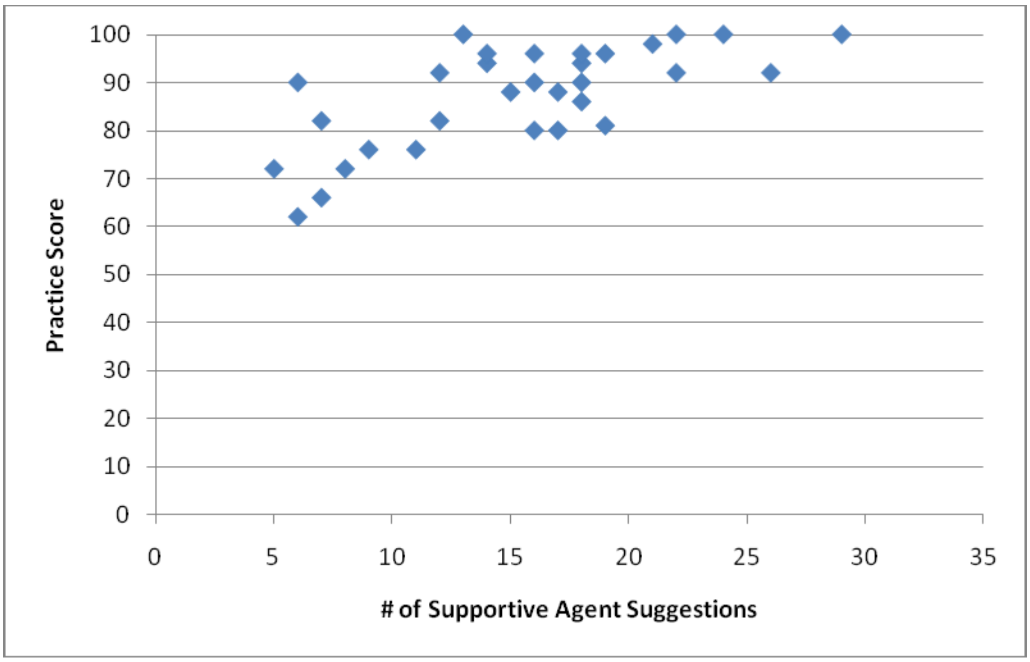

Figure 5.1 Scatter Diagram for Practice Score and Agent Suggestion

the correlation value is greater than 0.355 or less than -0.355, it can be concluded that the quite a bit higher, it can be concluded that it is not a chance finding and that the correlation is "statistically significant" (given the parameters of the test). The null hypothesis was rejected and the alternative was accepted.

5.2 Proposition 2

Practice part of the UML-ALF also has been experimented in Human Computer Interaction Laboratory. There is a single eye tracker (Tobii 1750) which collects data on where on the screen the user looks, how long a glance is, how many times the user looks at a certain point on the screen, and all the eye movements of the subject during the experiment is also monitored. Tobii Studio is software developed by the manufacturers of the eye tracker that transforms the information received from the reflectors, the infrared detector cameras into visual and digital data, records them and later provides various tools for analyzing the data saved. Five learners participated in the experiment. As three participants completed three of the practices, two participants completed just one practice. The observations during the experiments showed that supportive agents have a significant effect on completing the

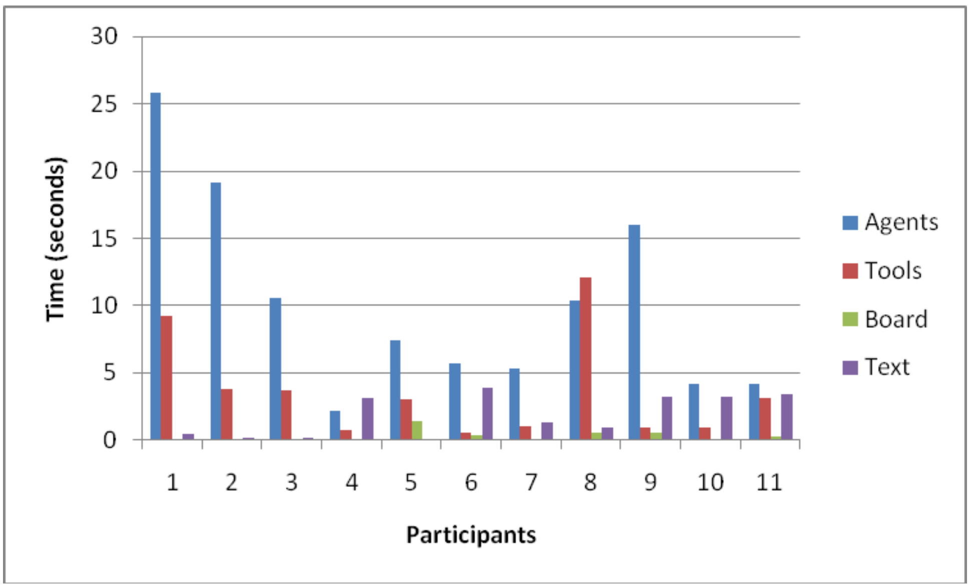

Figure 5.2 Time to First Fixation

practice and learning the subject. The heat map results of HCI lab experiments including mouse clicks are presented in Appendix D. In this map, the fixations are represented according to their time and number as a degrading pattern (red color represents the areas which are looked longer and most). By means of these output, at a first glance, the information about the areas which the participants look most is obtained. It is indicated that all participants were interested with supportive agents and they completed the practices successfully. Most of the participants' interaction data are approximately 80% valid, whereas participant 2 had about 55%. It is assumed that the reason for the shortfall of data gathering is that he was wearing glasses. And it is believed that because of this reason, there was not as much intensity on supportive agents as other participants on his heat map results.

Another evaluation criterion is the statistics of area of interests (AOI). The practice screen of UML-ALF is divided into four parts which are called as area of interest in Tobii Studio. First area is the right side of the screen where four agents exist. Second area is the left side of the screen that tools are presented. Third area is the board which is the main screen the user makes practice. Fourth area is the top of the screen where the text of practice exists.

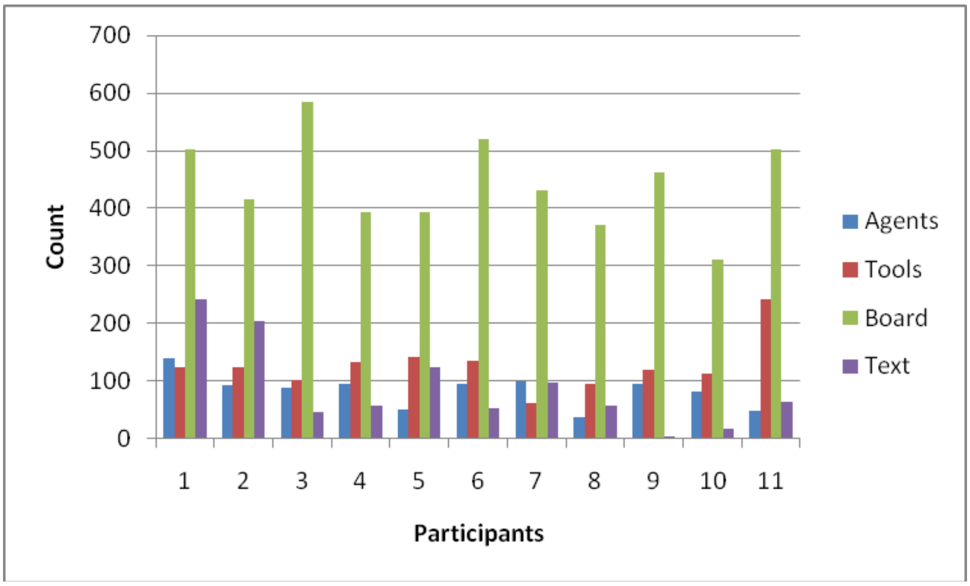

Figure 5.3 Fixation Count

Five filters were used to collect the statistics of area of interests: time to first fixation, fixation duration, fixation count, visit duration and visit count. Figure 5.2 shows the first fixation times of each areas. It is clearly seen that mostly participants first look at board area since it is the center of the screen. Then, learners focus on the practice text.

Following area is the tools area where they start their practice. Lastly, participants glance the agents. It is believed that this is the ordinary path participants would follow. When evaluating the results of statistics for agent area, blinking agents and their responses with short sentences should be considered as the duration of their fixation and visit parameters would not last long. Figure 5.3 presents the fixation parameters of the AOI. According to these figures, board is the area where participant most focused on. It is not expected that the agent area would be the most focused part among others, since it is a place where participants look for short periods. However, the result of fixation times shows that the agent part is used nearly as much as text and tools areas.

According to 5.4 which introduces the visit parameters of the AOI, the board area is again the most visited part. The result of these figures also confirms the results of fixation parameters. The visit duration and count values for the agent area is also as much as text and tools areas.

Mouse clicks are accepted as verification that participants get suggestion from supportive agents. Left mouse clicks on agent areas are illustrated in screenshots with mouse icons. (Appendix E)

Total fixation count, visit count and number of mouse clicks are shown in Figure 5.5. Average values of fixation counts, visit counts and number of mouse clicks are 89.09, 51 and 39.09 respectively. Fixation count implies number of times the participant fixates on AOI. Visit count is the number of visits within an AOI. The difference between fixation count and visit count is that when participant fixates on some point then in another point in same AOI, it is counted as 2 for fixation count and 1 for visit count. So, if visit counts and mouse clicks are equal, then it means that each time participant looked at agent area, he/she get suggestion from Supportive Agents. However, when participant looks at blinking agents and he/she might not click. According to the average results of HCI experiments (Figure 5.5), it is concluded that with a 77% probability, participants get help from Supportive Agents when they look at them.

As a consequence, it is accepted that participants are interacted with supportive agents due to the results of HCI experiments. Second proposition has already showed that the correlation of the participant's score with supportive agent's suggestion.

In practice part of the UML-ALF, some agents may give misleading hints which are common mistakes of other learners. In order to decide whether this methodology contributes to learning achievement of the user, a question was asked in evaluation survey. According to the result of the qualitative research (Figure 5.26) 56% of the participants agree with the idea of supportive agents that give misleading hints has positive effect on their learning process, whereas 19% of the participants reject the idea.

Another methodology used in practice part was tracking learner's movements. When learner makes same mistakes again and again, system thinks that there would be a problem

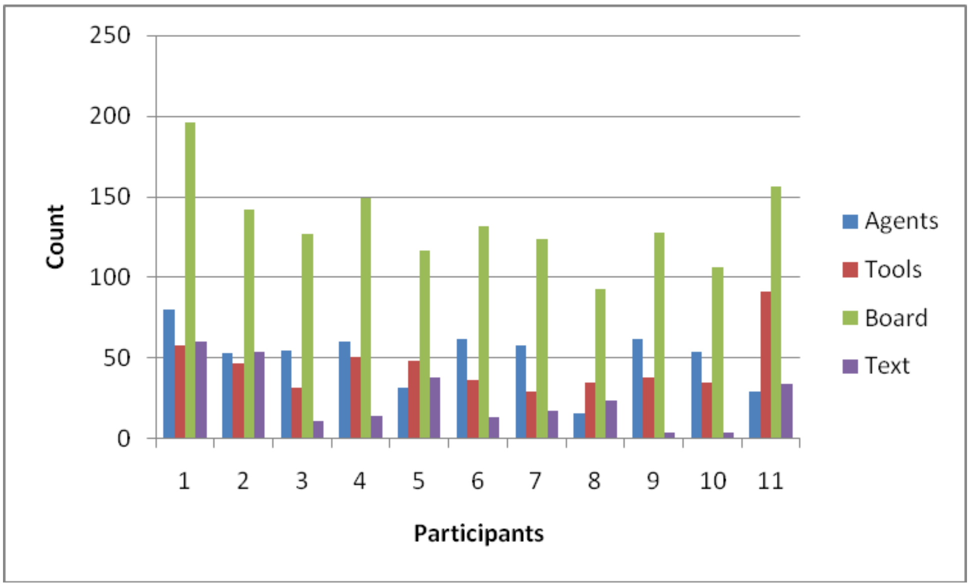

Figure 5.4 Visit Count

and asks a relevant question to the learner. If learner answers this question correctly, he/she goes on practice; otherwise a summary of the relevant topic is shown to the learner. Then, another question is asked. If learner still cannot find the correct answer, finally he/she is

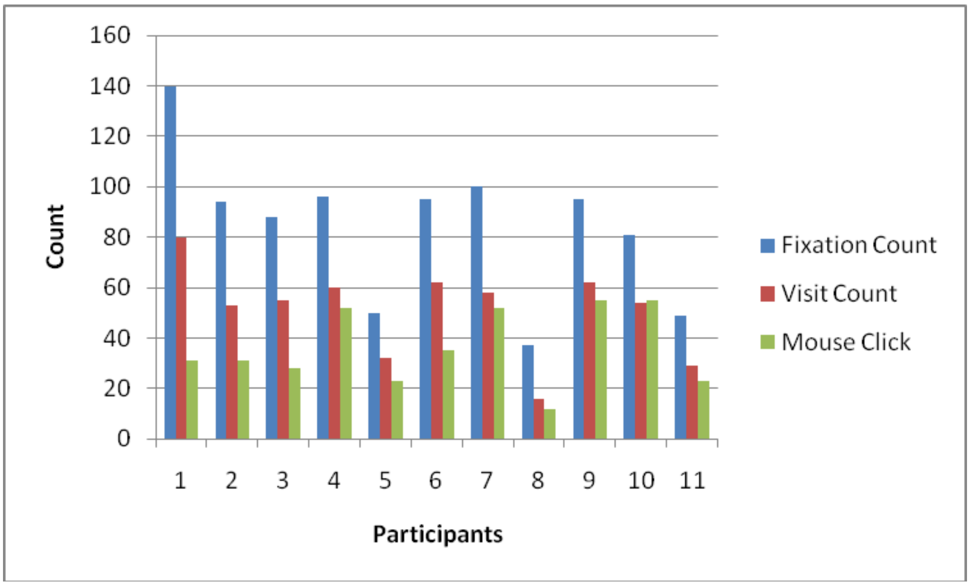

Figure 5.5 Total Fixation, Visit Count and Mouse Clicks for Agent Area

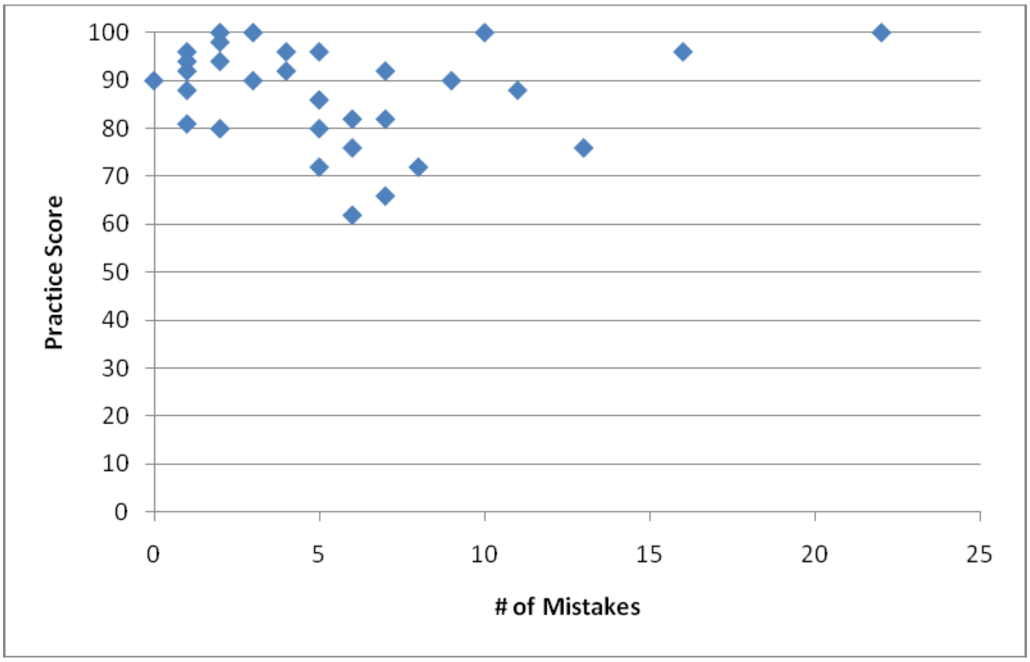

Figure 5.6 Scatter Diagram between Practice Score and Number of Mistakes

Table 5.3 Statistics for correlation between # of mistakes and Practice Score

Participants	# of mistakes (x)	score (y)	x*y	x*x	y*y
1	10	100	1000	100	10000
2	1	81	81	1	6561
3	4	92	368	16	8464
4	13	76	988	169	5776
5	7	92	644	49	8464
6	2	100	200	4	10000
7	16	96	1536	256	9216
8	2	94	188	4	8836
9	0	90	0	0	8100
10	3	100	300	9	10000
11	1	88	88	1	7744
12	7	66	462	49	4356
13	5	80	400	25	6400
14	6	62	372	36	3844
15	2	80	160	4	6400
16	1	96	96	1	9216
17	8	72	576	64	5184
18	1	92	92	1	8464
19	4	96	384	16	9216
20	6	82	492	36	6724
21	3	90	270	9	8100
22	2	98	196	4	9604
23	5	86	430	25	7396
24	6	76	456	36	5776
25	5	72	360	25	5184
26	9	90	810	81	8100
27	1	94	94	1	8836
28	22	100	2200	484	10000
29	5	96	480	25	9216
30	7	82	574	49	6724
31	11	88	968	121	7744
Sum=	175	2707	15265	1701	7327849

redirected to the course of the subject. The correlation between two variables, the usage of supportive agents by the learner and their practice score was based on. In order to evaluate the methodology that tracks user movements, a correlation between number of mistakes and practice score is examined. The hypothesis is that how much learner made a mistake affects their practice score. Participants that made use the benefits of the intelligent techniques of UML-ALF has produced totally 31 practice results in 3 practices (Class Diagrams, Use Case Diagrams and Activity Diagrams). According to the result of Table 5.7, r is calculated as -0.0002. The result of r and Figure 5.6 show that there is no correlation between variables practice score and number of mistakes. However, according to the Table 5.7, it is also seen that participants that made mistakes also get high scores from practices. It can be concluded that the methodology of tracking users has the ability of teaching subject to the inexperienced users and increasing their learning achievement.

Also, a question was asked about tracking user movements in evaluation survey. According to the qualitative research (Figure 5.24) 47% of the participants agree with the idea of tracking user movements and redirecting them to content materials, whereas 44% of the participants reject the idea. The reason for the high percentage of disagreement might not be related with the failure of the methodology, on the contrary it can be related to exhausting effects of redirecting user from practice screen.

5.3 Proposition 3

Third proposition is introduced as intelligent agents and computer based tutors can give effective feedback to the learners which is close to the performance of human tutors. The result of this proposition is determined by the evaluation surveys which were filled out by participants after experiencing UML-ALF. According to the results of these surveys, the feedback which is given by intelligent agent and computer based tutors in this study is satisfactory; however it is still not so close to human tutors. (Figure 5.20)

Since it is a relative parameter that feedback is effective or not, in order to make more accurate and quantitative decisions, in future works, different learning platforms can be compared with a larger sample of participants.

5.4 Survey Results

Results of each question in evaluation survey [Appendix C] are presented in pie charts. The results of evaluation survey show that participants find the content material sufficient in terms of information and visuality. It is also verified that adaptive techniques that were implemented in study part of UML-ALF was helpful. Most of the participants find the learning path convenient which is assigned by the system. It is stated that supportive agents' help was valuable. They agree that practice screens help to test and improve their knowledge about the subject. However, according to the participants, the feedback generated in practice part was not enough. As a consequence, they think that system is successful in teaching UML.

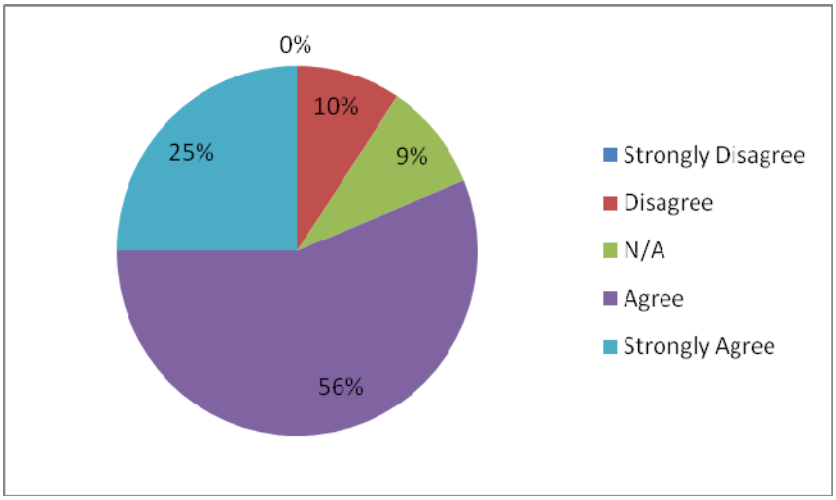

Figure 5.7 Result of Question 1 in Evaluation Survey

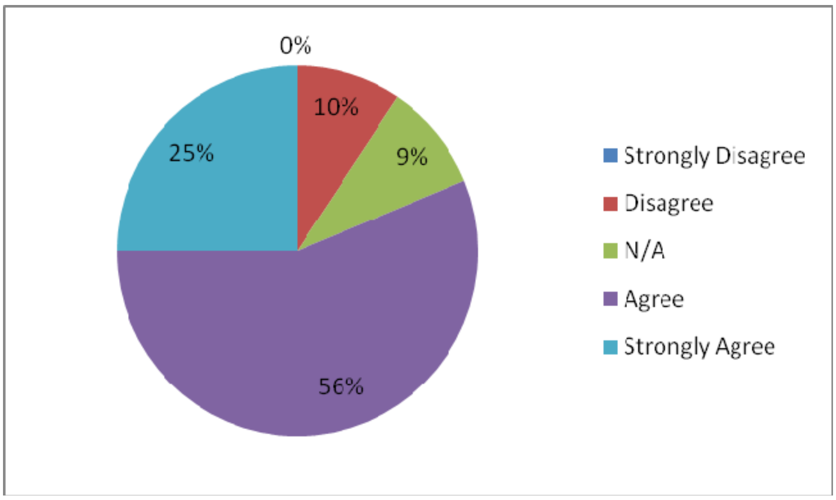

Figure 5.8 Result of Question 2 in Evaluation Survey

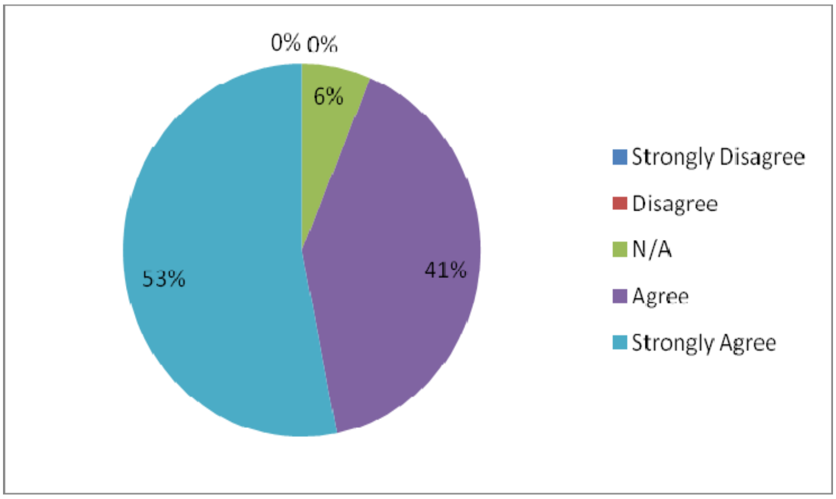

Figure 5.9 Result of Question 3 in Evaluation Survey

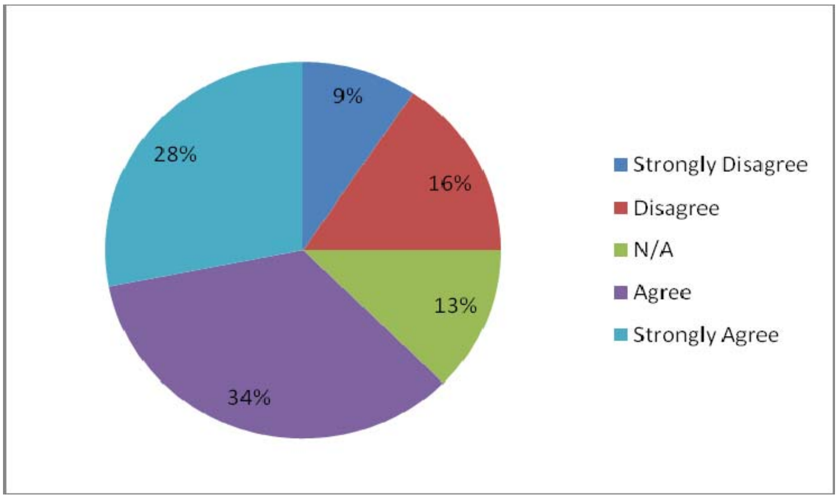

Figure 5.10 Result of Question 4 in Evaluation Survey

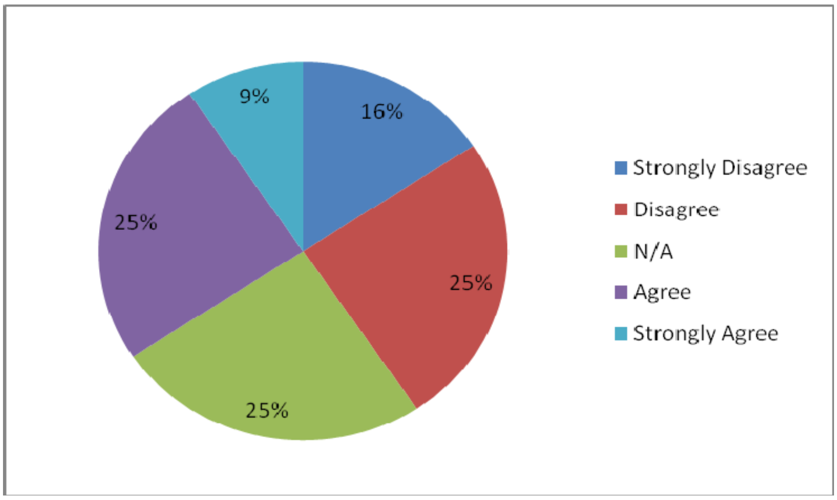

Figure 5.11 Result of Question 5 in Evaluation Survey

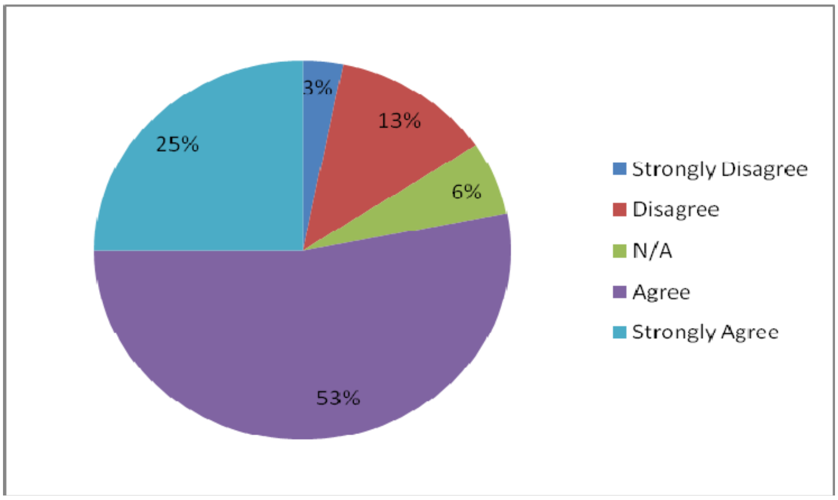

Figure 5.12 Result of Question 6 in Evaluation Survey

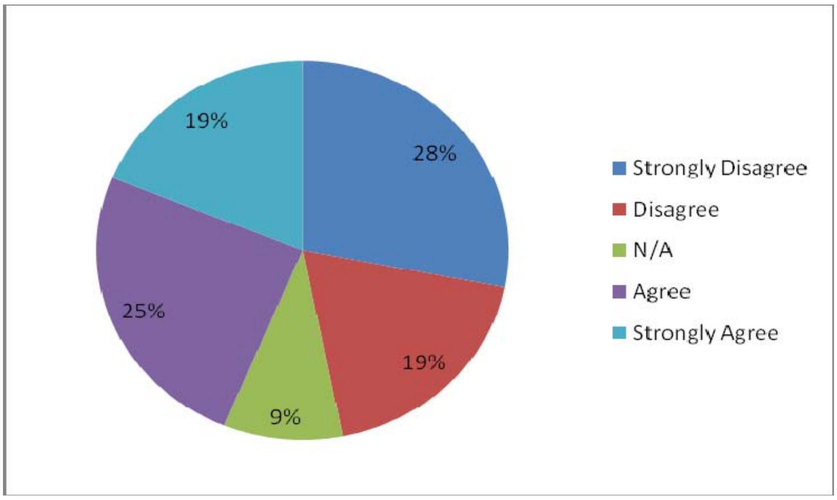

Figure 5.13 Result of Question 7 in Evaluation Survey

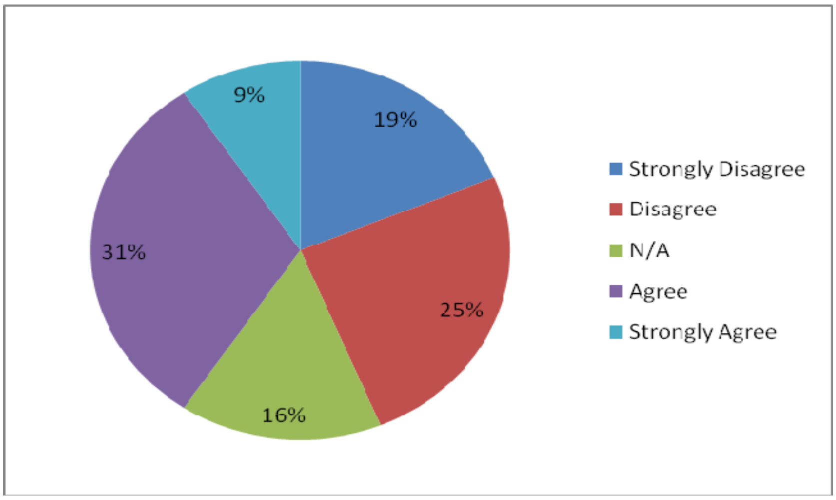

Figure 5.14 Result of Question 8 in Evaluation Survey

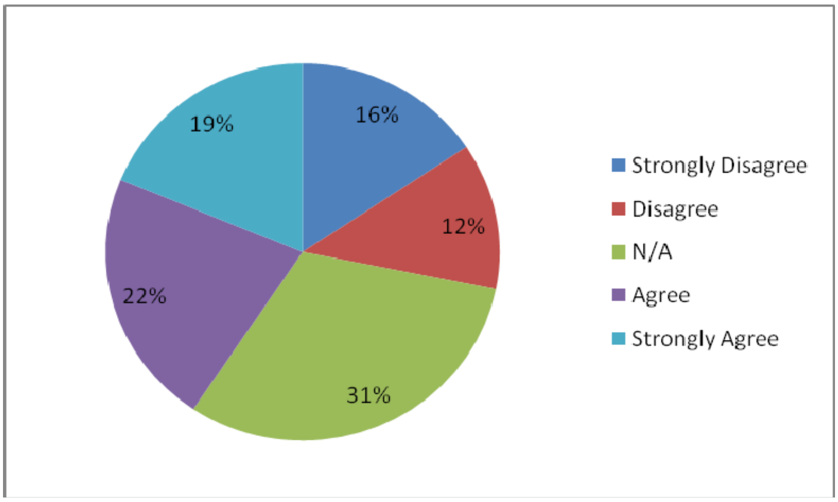

Figure 5.15 Result of Question 9 in Evaluation Survey

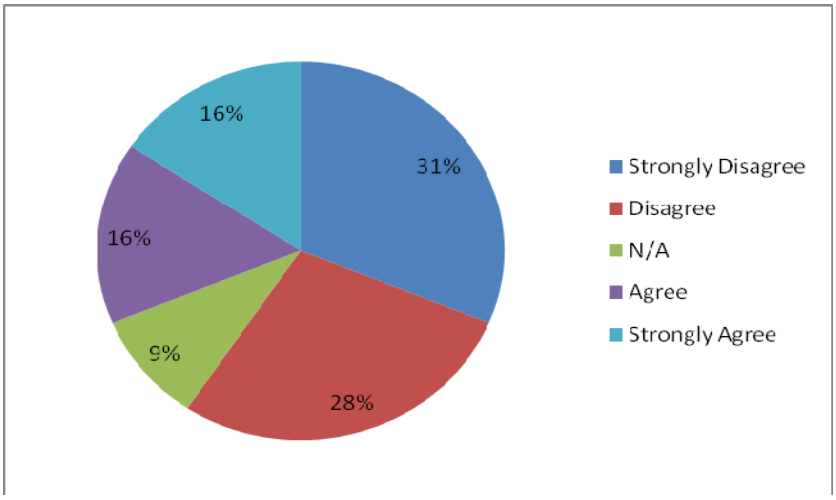

Figure 5.16 Result of Question 10 in Evaluation Survey

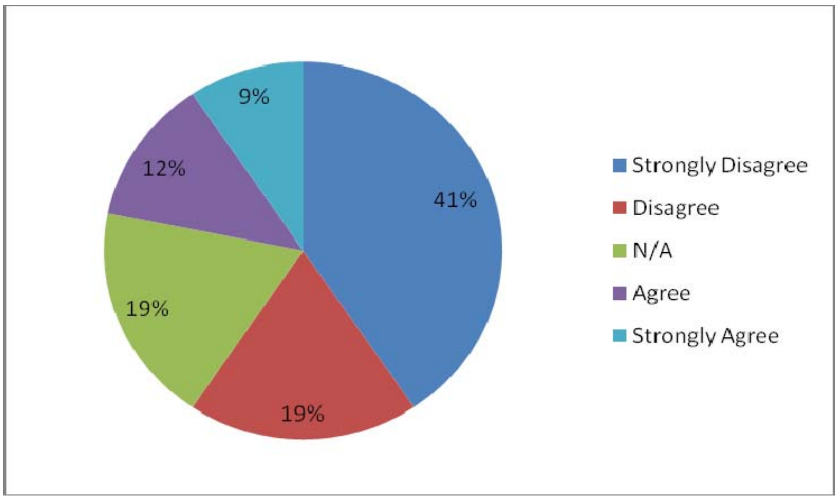

Figure 5.17 Result of Question 11 in Evaluation Survey

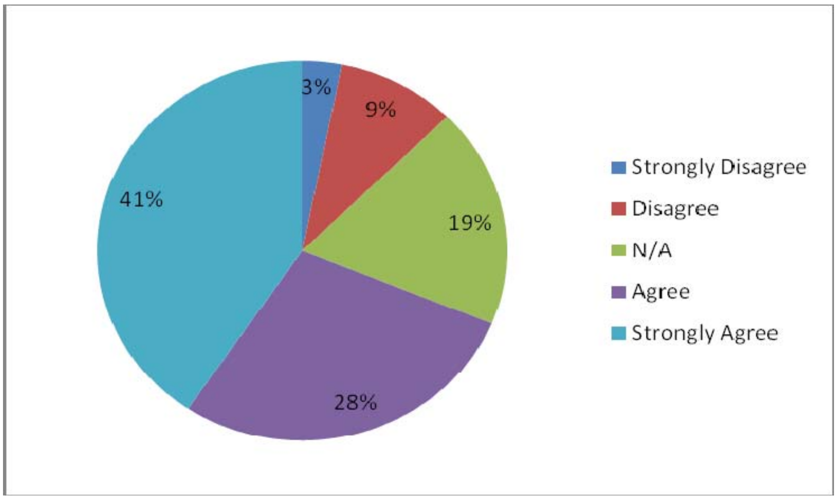

Figure 5.18 Result of Question 12 in Evaluation Survey

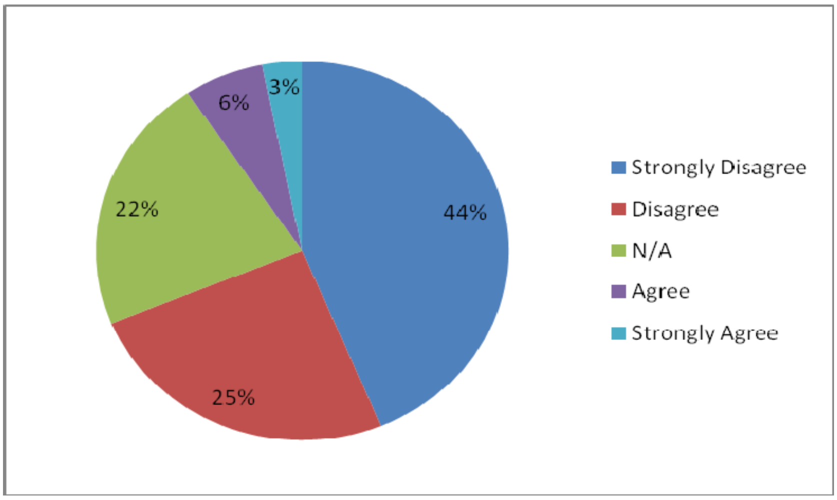

Figure 5.19 Result of Question 13 in Evaluation Survey

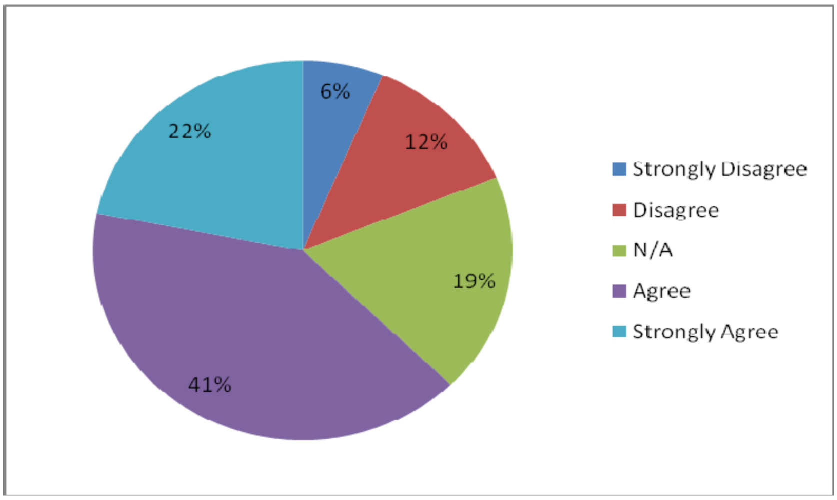

Figure 5.20 Result of Question 14 in Evaluation Survey

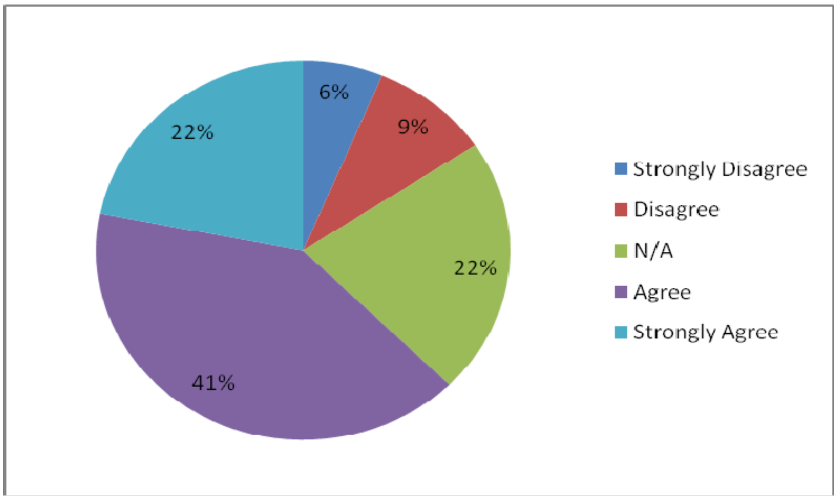

Figure 5.21 Result of Question 15 in Evaluation Survey

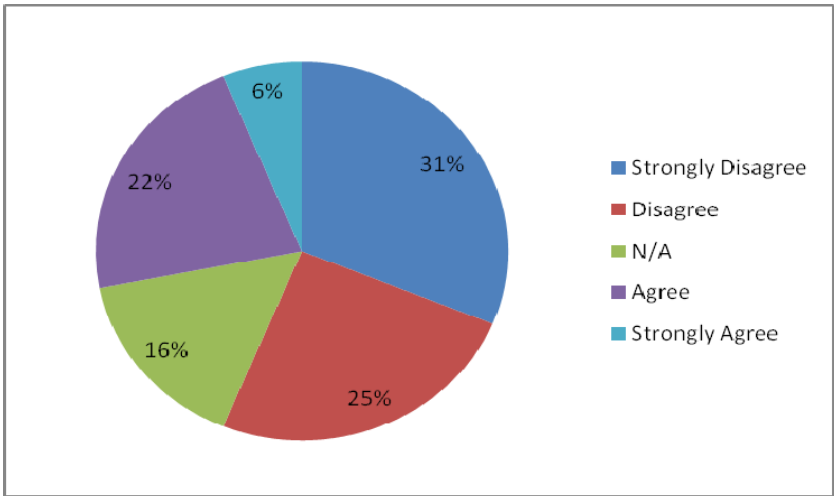

Figure 5.22 Result of Question 16 in Evaluation Survey

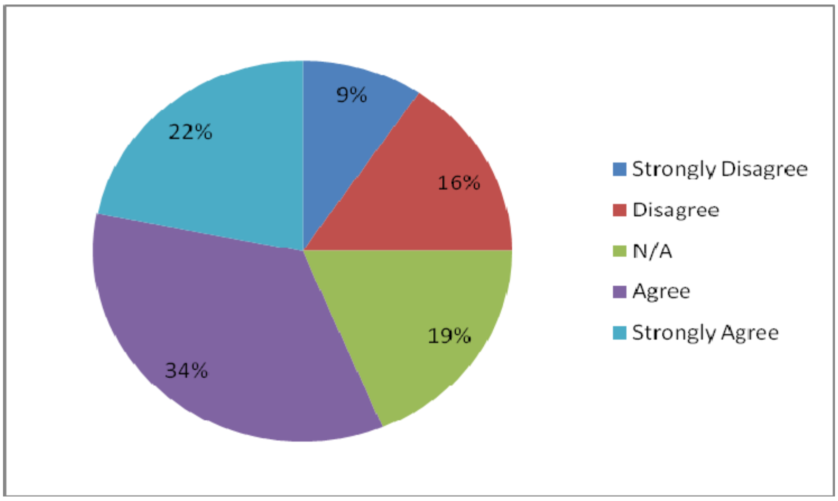

Figure 5.23 Result of Question 17 in Evaluation Survey

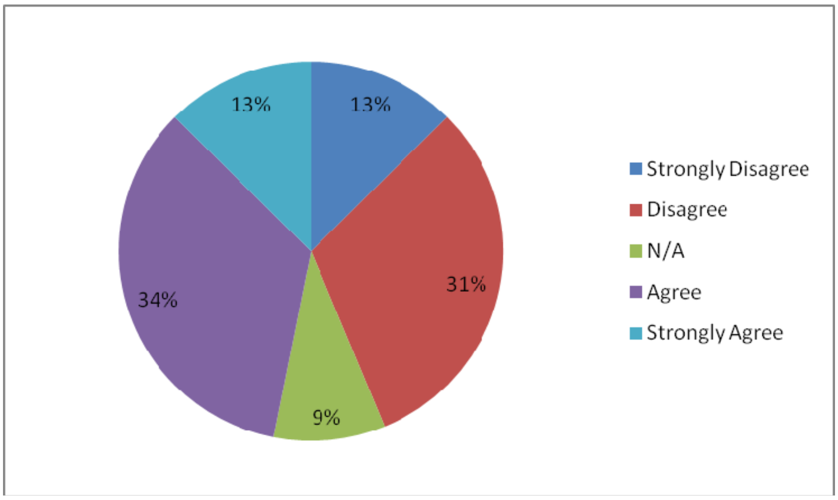

Figure 5.24 Result of Question 18 in Evaluation Survey

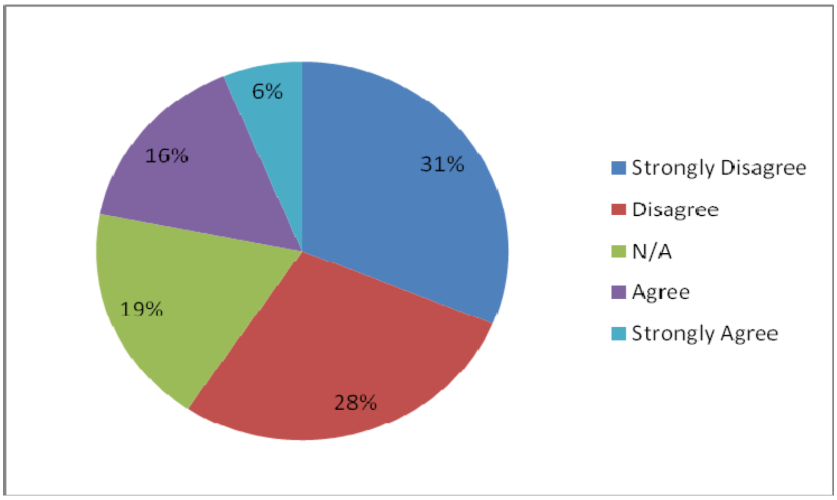

Figure 5.25 Result of Question 19 in Evaluation Survey

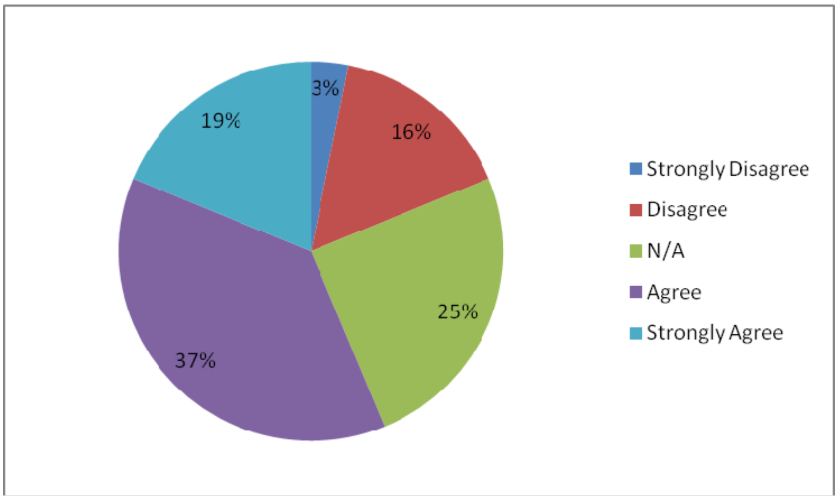

Figure 5.26 Result of Question 20 in Evaluation Survey

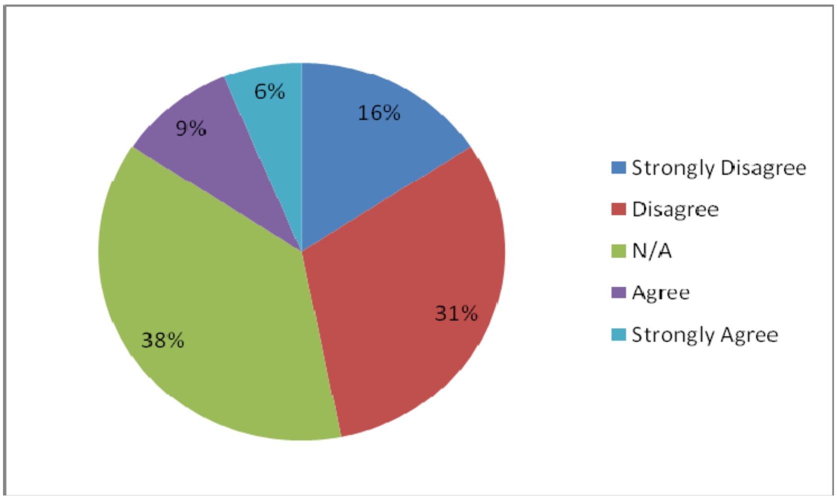

Figure 5.27 Result of Question 21 in Evaluation Survey

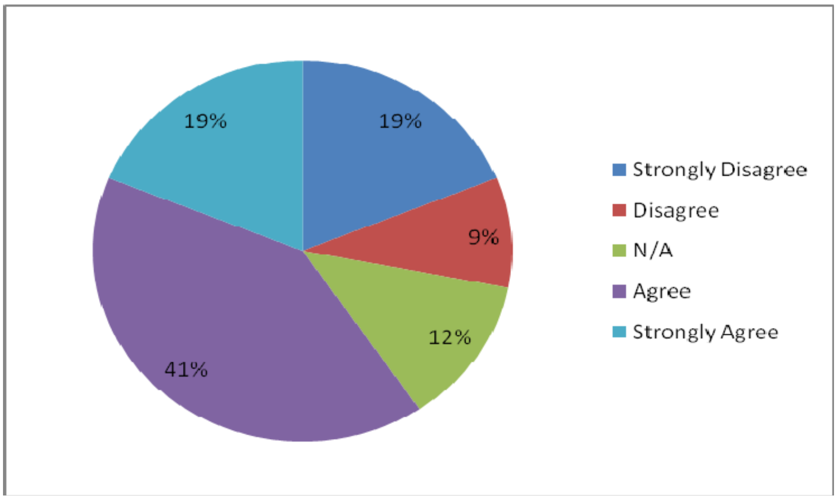

Figure 5.28 Result of Question 22 in Evaluation Survey

CHAPTER 6

DISCUSSION

In order to assess that UML-ALF has accomplished its objectives, we followed an experimental procedure. Three different features of this framework were considered during the experiment. Those features are contribution of supportive agents, interaction between SAs and learners, and advanced feedback mechanism.

6.1 The Contribution of Supportive Agents

The correlation coefficient r = 0.69 shows that there is a positive correlation between variables practice score and number of agent suggestion. This means, as the participants benefit from supportive agents, they get higher scores. With the significant test, it is concluded that the value is not a chance finding and that the correlation is "statistically significant". The null hypothesis was rejected and the alternative was accepted.

The participants of the study were examined in terms of their professions. According to the results of survey [Appendix B] participants are gathered in two groups. These groups are named as Experienced and Inexperienced. People in experienced group has advanced knowledge about UML, they took a UML related course in their education, use the benefits of UML diagrams in their current work. On the contrary, participants in inexperienced group are lack of UML knowledge or little if any, however they are interested in subject. The correlation coefficient of inexperienced group was 0.81 whereas it is 0.62 for experienced group. This result implies that there is more linear relationship between practice score and number of agent suggestion for inexperienced group than experienced group. It can be concluded that inexperience group had much to learn and benefit from supportive agents more than experienced group.

A similar work which is an adaptive educational hypermedia system called MATHEMA (Papadimitriou et al., 2009) also supports curriculum sequencing, adaptive presentation, adaptive navigation and interactive problem solving techniques. It is stated that an interactive tutor cannot be implemented in their work. The agents sleeps from one help request to another. On the contrary, in UML-ALF supportive agents response to each user move during their learning period. Since domains and evaluation methods are different for both works, the comparison of the success of frameworks cannot be precise.

6.2 The Interaction between Supportive Agents and Learners

6.2.1 HCI experiments

Practice part of the UML-ALF also has been experimented in Human Computer Interaction Laboratory. Mouse clicks are accepted as verification that participants get suggestion from supportive agents. Left mouse clicks on agent areas are illustrated in screenshots with mouse icons (Appendix E). Average values of fixation counts, visit counts and number of mouse clicks are 89.09, 51 and 39.09 respectively. Fixation count implies number of times the participant fixates on AOI. Visit count is the number of visits within an AOI. The difference between fixation count and visit count is that when participant fixates on some point then in another point in same AOI, it is counted as 2 for fixation count and 1 for visit count. So, if visit counts and mouse clicks are equal, then it means that each time participant looked at agent area, he/she get suggestion from Supportive Agents. However, when participant looks at blinking agents and he/she might not click. According to the average results of HCI experiments, it is concluded that with a 77% probability, participants get help from Supportive Agents when they look at them. It is a high usage rate for SAs. Therefore, correlation coefficient result shows that supportive agents play important role for increasing learning achievement of the user. In addition to this, HCI experiment results verify this conclusion with a high usage rate.

6.2.2 Suggesting Common Mistakes as Misleading Hints

The communication protocol of the agents and building the practice resources are based on the study of Mengelle et al. (1998). However, since domains of these studies are different, techniques for building the resource are also different. Language agent is responsible for building and interpreting the resource in UML-ALF. Mengelle et al. (1998) focus on modeling of pedagogical expertise and actors' learning abilities whereas in our study the purpose was to implement these methodologies. The idea of generating misleading hints is originated from Mengelle's work. But this idea is developed by giving common mistakes of other users as misleading hints in our work. Remolina et al. (2009) introduce

Automated Role Players (ARPs) in their simulation-based Intelligent Tutoring System for training Tactical Action Officers (TAOs). ARPs also tend to make mistakes intentionally. TAO should notice the mistake and take action. If the TAO does not respond, ARPs or ARP Captain intervene and correct the situation themselves. This is a similar approach as supportive agents give misleading hints. At the end, tutor agent corrects the mistake as ARP Captain does. Learning from mistakes is one of the learning methodologies. Trying to mislead learner with a common mistake is also used in Mengelle et al. (1998). The benefit of learning by making mistakes is also discussed in (Bak & Chialvo, 1999; Dweck, 2007; Izzo & Mazzone, 2008).

According to the result of the evaluation survey 56% of the participants agree with the idea of supportive agents that give misleading hints has positive effect on their learning process, whereas 19% of the participants reject the idea.

6.2.3 Tracking Learner's Movements

Another methodology used in practice part was tracking learner's movements. When learner makes same mistakes again and again, system thinks that there would be a problem
In order to evaluate the methodology that tracks user movements, a correlation between number of mistakes and practice score is examined. The hypothesis is that how much learner made a mistake affects their practice score. However, the correlation coefficient (0.0002) and Figure 5.6 show that there is no correlation between variables practice score and number of mistakes. On the other hand, according to the practice results, it is also seen that participants that made mistakes also get high scores from practices. It can be concluded that the methodology of tracking users has the ability of teaching subject to the inexperienced users and increasing their learning achievement.

According to the survey results (Figure 5.24) 47% of the participants agree with the idea of tracking user movements and redirecting them to content materials, whereas 44% of the participants reject the idea. The reason for the high percentage of disagreement might not be related with the failure of the methodology, on the contrary it can be related to exhausting effects of redirecting user from practice screen.

6.3 Effective Feedback

Fossati et al. (2009) develop an ITS system which helps students to understand the subject of Linked Lists easily which is an important topic in Computer Science. It is believed that most of the computer science students have difficulties in learning this subject. According to the feedback of students, system is considered interesting, useful and its performance is getting closer to the performance of human tutors. In UML-ALF intelligent agents and computer based tutors also can give effective feedback to the learners which is close to the performance of human tutors. The result of this proposition is determined by the evaluation surveys which were filled out by participants after experiencing UML-ALF. According to the results of these surveys, the feedback which is given by intelligent agent and computer based tutors in this study is satisfactory; however it is still not so close to human tutors.

Since it is a relative parameter that feedback is effective or not, in order to make more accurate and quantitative decisions, in future works, different learning platforms can be compared with a larger sample of participants.

Kenny and Pahl (2009) present an interactive training and learning environment. It has an automated and adaptive tutoring system at the core. This system supports the Structured Query Language (SQL) part of a database courseware system. They evaluate their system with surveys. Forty percent agreed that their feedback mechanism is helpful whereas feedback mechanism of UML-ALF is found 63% useful(Figure 5.20). Over twofifths of students agreed that their guidance feature is useful. The contribution of supportive agents is discussed in Part 6.1 and 6.2. Also in our evaluation survey 69% of the participants agreed that supportive agents are very helpful.

CHAPTER 7

CONCLUSION and FUTURE WORK

Interactive and adaptive learning environments differs substantially from traditional instructor-oriented, classroom-based teaching.This study describes an agent-based adaptive learning framework whose goal is to implement effective tutoring system with the help of Artificial Intelligence (AI) techniques and cognitive didactic methods into Adaptive Educational Hypermedia Systems (AEHS). There are three main goals of this study. First goal is to explore how supportive agents affect student's learning achievement in distance learning. Second goal is to examine the interaction between supportive agents and learners with the help of experiments in Human Computer Interaction laboratories and system analysis. The effects of the methodology that agents give misleading hints which are common mistakes of other learners are also investigated. Last goal is to deliver effective feedback to students both from IAs and tutors.

Learning is usually divided into two categories – knowledge acquisition and skills training. When learners are actively engaged, they reach a higher level of understanding while the types are different. There should be a high level of knowledge-level and skills-level interaction between the student and a tutoring system. Scaffolding and feedback is the most important feature of the apprenticeship theory. (Kenny & Pahl, 2009)

It is expected that this study determines how adaptive hypermedia systems and intelligent agents contribute on distance education. Besides, this study gathers the information of which methodologies of adaptive and intelligent techniques have positive effects on learning, which learning path the participants follow, how they react to the learning system, with the help of analysing data of intelligent agents, face-to-face trials and observations of experiments in Human Computer Interaction Labs.

Considering the experience gained during this study and the findings of this study, some recommendations for further research are provided. This study is conducted based on providing an adaptive learning framework for all domains. Thus, this framework might be used for different domains or multi-domain implementations in a future work.This study is implemented on UML domain with semi-professional content materials and small sample group. In a future work for better results, more professional materials – contents, exercises, practices – and broader sample group might be used.

REFERENCES

Bell, A.E. (2004). Death by UML Fever, *ACM Queue*, 2(1), 72-80.

Bowman, M., Lopez Jr., A. M., Donlon, J. & Tecuci, G. (2001). Teaching Intelligent Agents: Software Design Methodology. *CrossTalk - the Journal of the Defense Software Engineering*, 14(6), 10-14, Retrieved December 27, 2009 from http://www.stsc.hill.af.mil/crossTalk/2001/06/bowman.html

Brusilovsky, P. (2000). Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education. *Lecture Notes in Computer Science, Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, 1839, 156-169.

Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Instruction*, 11(1-2), 87-110.

Brusilovsky, P. & Peylo, C. (2003). Adaptive and Intelligent Web-Based Educational Systems. *J. Artificial Intelligence in Education*, 13, 156-169.

Cabukovski, V.E. (2006). An Agent-Based Testing Subsystem in an E-Learning Environment. *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, 622-625.

Chen, T. (2008). Mutual Model of Multiagent Process in E-learning Architecture. *ISISE '08, International Symposium on Information Science and Engineering*, 2, 154-158.

Chialvo, D. R. & Bak, P. (1999). Learning from mistakes. *Neuroscience,* 90(4), 1137-1148.

Dweck, C.S. (2007). Perils and Promise of Praise. *Educational Leadership – Early Intervention at Every Age*, 65(2), 34-39.

Faria, L., Silva, A., Vale, Z. & Marques, A. (2009). Training Control Centers' Operators in Incident Diagnosis and Power Restoration Using Intelligent Tutoring Systems. *IEEE Transactions on Learning Technologies*, 2(2), 135-147.

Feng, M., Heffernan, N.T., Heffernan, C. & Mani, M. (2009). Using Mixed-Effects Modeling to Analyze Different Grain-Sized Skill Models in an Intelligent Tutoring System. *IEEE Transactions on Learning Technologies*, 2(2), 79-92.

Fossati, D., Eugenio, B.D., Brown, C.W., Ohlsson, S., Cosejo, D.G. & Chen, L. (2009). Supporting Computer Science Curriculum Exploring and Learning Linked Lists with iList. *IEEE Transactions on Learning Technologies*, 2(2), 107-120.

Hambleton, R. K. & Swaminathan, H. (1985). *Item Response Theory*. Boston, MA: Kluwer-Nijhoff.

Harris, T. (2009). *yUML, yUML beta v0.18*. Retrieved December 27, 2009 from http://yuml.me

Hong, C.M., Chen, C.M., Chang, M.H. & Chen, S.C. (2007). Intelligent Web-based Tutoring System with Personalized Learning Path Guidance. *Seventh IEEE International Conference on Advanced Learning Technologies,* 512-516.

Huffman, S.B. (1994). Instructable Autonomous Agents. *PhD Thesis, University of Michigan, Dept of Electrical Engineering and Computer Science*.

IMS (2001). IMS Learner Information Packaging. Retrieved June 6, 2010 from http://www.imsproject.org/profiles/lipbest01a.html

Izzo, M. F. & Mazzone, G. (2008). Learning by Doing Mistakes Improving ICT Systems Through the Evaluation of Application Mistakes. *Interdisciplinary Aspects of Information Systems Studies, Physica-Verlag HD*, 123-129.

Jin, L. L. & Ming, H. C. (2008). An Intelligent Agent for Personalized E-Learning, *The Eighth International Conference on Intelligent Systems Design and Applications*, 1, 27-31.

Keller, F. S. & Sherman, J. G. (1974). *The Keller Plan Handbook*. Menlo Park, CA: W. A. Benjamin.

Kenny, C. and Pahl, C. (2009). Intelligent and adaptive tutoring for active learning and training environments, *Interactive Learning Environments*, 17(2), 181 — 195

Kızılkaya, G. and Aşkar, P. (2008). The effect of an pedagogical agent on the students' science achievement, *Interactive Technology and Smart Education*, 5(4), 208-216.

Kolb, D.A. (1984). *Experiential Learning: Experience as the Source of Learning and Development.* Prentice-Hall, Inc.

Kolen, M. J., Hanson, B. A. & Brennan, R. L. (1992). *Conditional standard errors of measurement for scale scores*. J. Educ. Measur., 29, 285–307.

Lynch, G., Palmiter, S. & Tilt, C. (1999). The Max Model: A Standard Web Site User Model. *5th Conference on Human Factors & the Web*

Mengelle, T., de Léan, C. & Frasson, C. (1998). Teaching and Learning with Intelligent Agents: Actors. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems, Heidelberg, Germany: Springer-Verlag*, 1452, 284 – 293,

Papadimitriou, A., Grigoriadou, M. & Gyftodimos, G. (2009). Interactive Problem Solving Support in the Adaptive Educational Hypermedia System MATHEMA. *IEEE Transactions on Learning Technologies*, 2(2), 93-106.

Patton, M.Q. (2002). *Qualitative Research and Evaluation Methods (3rd ed.)*. Thousand Oaks: Sage Publications

Remolina, E., Ramachandran, S., Stottler, R. & Davis, A. (2009). Rehearsing Naval Tactical Situations Using Simulated Teammates and an Automated Tutor. *IEEE Transactions on Learning Technologies,* 2(2), 148-156.

Russell, S. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach (AIMA)* (2^{nd} ed.). Prentice Hall, New Jersey.

Scott, W. L., Beichner, R. J., Titus, A. & Martin, L. (2000). *Education research using Web-based assessment systems*. J. Res. Comput. Educ., 33, 28–43.

Shih, T. K., Lin, N. H., Change, H. (2003). *An intelligent e-learning system with authoring and assessment mechanism*. IEEE Int. Conf. Advanced Information Networking Applications, 782–787.

Somyürek, S. (2009). Adaptive Learning Environments: A New Paradigm in Educational Hypermedia Design. *Bilişim Teknolojileri Dergisi,* 2(1), 29-38.

Tecuci, G. & Hieb, M.R. (1996). Teaching intelligent Agents: the Disciple Approach. *International Journal of Human-Computer Interaction*, 8(3), 259-285, 1996.

Uhr, L. (1969). Teaching machine programs that generate problems as a function of interaction with students. *ACM Annual Conference/Annual Meeting Proceedings of the 1969 24th national conference*, 125 – 134.

APPENDIX A

SCENARIOS of BUILDING RESOURCES

[U] User, [DA] Diagnostic Agent, [LA] Language Agent, [FM] Feedback Mechanism, [SA] Supportive Agent

[SA1], [SA2] are helping agents whereas [SA3], [SA4] are misleading ones.

[SA2], [SA4] are forced to response.

[DA]: - checkElement(38)

Rule ID	*Rule Syntax*	*Practice ID*	*Diagram Type*	*Element Type*	*Rule Owner*
38	*[Company]+1->*[Employee]*	*1*	*1(class-diagram)*	*5 (relations)*	*7(student)*

- generateReaction(38)

Rule ID	*Rule Syntax*	*Practice ID*	*Diagram Type*	*Element Type*	*Rule Owner*
39	*[Company]-0..*>[Employee]*	*1*	*1(class-diagram)*	*5 (relations)*	*1(DA)*
40	*(Company)<(Employee)*	*1*	*2(use-case dia.)*	*5 (relations)*	*1(DA)*
41	*The relationship between those subjects is not 'Composition'.*	*1*	*-*	*9(general hint)*	*1(DA)*

- send to SAs

[SA1]	*[SA2]*	*[SA3]*	*[SA4]*
- evaluateReactions() *- 39:unknown* *- 40:false* *- 41:false* *- selects no response*	*- evaluateReactions()* *- 39:unknown* *- 40:false* *- 41:false* *- selects 39*	*- evaluateReactions()* *- 39:true* *- 40:false* *- 41:unknown* *- selects 40*	*- evaluateReactions()* *- 39:true* *- 40:true* *- 41:unknown* *- selects 41*

Scenarios

[U] User, [DA] Diagnostic Agent, [LA] Language Agent, [FM] Feedback Mechanism, [SA] Supportive Agent

Scenario 1

[S]: - Selects UML diagram type (e.g. Class-Diagram)

- Selects element type (e.g. Class)
- Enters name (e.g. Employee)
- sends information to LA

[LA]: - creates rule package

Rule ID	*Rule Syntax*	*Practice ID*	*Diagram Type*	*Element Type*	*Rule Owner*
25	*[Employee]*	*1*	*1 (class-diagram)*	*3 (class)*	*7 (student)*

- send package to DA

[DA]: - checkElement(25) ---> OK!

- generateReaction(25)
- send to SAs

[SAs]: - decideReaction()

Scenario 2

[S]: - Selects UML diagram type (e.g. Class-Diagram)

- Selects element type (e.g. Class)
- Enters name (e.g. User)
- sends information to LA

[LA]: - creates rule package

Rule ID	*Rule Syntax*	*Practice ID*	*Diagram Type*	*Element Type*	*Rule Owner*
26	*[User]*	*1*	*1 (class-diagram)*	*3 (class)*	*7 (student)*

- send package to DA

[DA]: - checkElement(26) ---> Fail!

- generateReaction(26)
- send to SAs

[SAs]: - decideReaction()

Scenario 3

[S]: - Asks help from SA2

[SA2]: - getReactions(1) (sending practice ID)

[DA]: - loadPractice(1)

- generateReaction(1)

- send to SA2

[SAs]: - decideReaction()

Scenario 4

[S]: - finishPractice()

[DA]: - checkPractice(1)

---> OK!

[DA]: - send results to FM

[FM]: - response positive feedback

---> Fail!

[DA]: - diagnosePractice(1)

[DA]: - send results to FM

[FM]: - response intelligent negative feedback

APPENDIX B

SURVEY FOR DETERMINING EXPERIMENT GROUPS

1. Uzmanlık Alanınız : ………………………………………

   □Bilgisayar Bilimleri

   □Bilgi Teknolojileri

   □Eğitim

   □Diğer

2. Unified Modeling Language (UML) ile ilgili ders aldınız mı?

   □ Evet

   □ Hayır

3. Unified Modeling Language (UML) iş yaşamınızda kullanıyor musunuz?

   □ Evet

   □ Hayır

4. Unified Modeling Language (UML) bilgi seviyenizi nasıl tanımlarsınız?

   □Hiç

   □Az

   □Orta

   □İleri

5. Unified Modeling Language (UML) ile ne kadar ilgileniyorsunuz?

   □Hiç

   □Az

   □Orta

   □Çok

APPENDIX C

EVALUATION SURVEY

Çoktan Seçmeli Sorular

1. Eğitimlerin içeriği bilgi bakımından yeterliydi.
   - o Kesinlikle katılmıyorum
   - o Katılmıyorum
   - o Fikrim yok
   - o Katılıyorum
   - o Kesinlikle katılıyorum
2. Eğitimlerin içeriği görsellik bakımından yeterliydi.
   - o Kesinlikle katılmıyorum
   - o Katılmıyorum
   - o Fikrim yok
   - o Katılıyorum
   - o Kesinlikle katılıyorum
3. Eğitimleri hangi sırada almam gerektiği açıktı.
   - o Kesinlikle katılmıyorum
   - o Katılmıyorum
   - o Fikrim yok
   - o Katılıyorum
   - o Kesinlikle katılıyorum
4. Eğitimlerin öğrenmemde katkısı olduğunu düşünüyorum.
   - o Kesinlikle katılmıyorum
   - o Katılmıyorum
   - o Fikrim yok
   - o Katılıyorum
   - o Kesinlikle katılıyorum

5. Bana sunulan eğitim yolu benim için uygundu.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
6. Eğitimler, eğitim sonrasında karşıma çıkan testi yapmamda faydalıydı.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
7. Eğitimlerin içeriğinde zaten hakim olduğum konular bulunuyordu.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
8. Eğitimler gereksiz ve fazla bilgi içeriyordu.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
9. Eğitimleri kendi tercih ettiğim sırada almak isterdim.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
10. Eğitim ekranlarında kullanıcıyı yönlendirme yetersizdi.
    - Kesinlikle katılmıyorum
    - Katılmıyorum

- Fikrim yok
- Katılıyorum
- Kesinlikle katılıyorum

11. Eğitim ekranları kullanılabilirlik açısından yetersizdi.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
12. Alıştırma kısmında ajanların yardımları çok faydalıydı.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
13. Alıştırma kısmında ajanların yorumları beni olumsuz etkiledi.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
14. Alıştırma ekranlarındaki öğretmenin geri bildirimleri, gerçek öğretmenin geri bildirimlerine yakındı.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum
15. Alıştırma ekranları bilgimi sınamama ve geliştirmeme yardımcı oldu.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum

16. Alıştırma ekranları kullanılabilirlik açısından yetersizdi.
   - Kesinlikle katılmıyorum
   - Katılmıyorum
   - Fikrim yok
   - Katılıyorum
   - Kesinlikle katılıyorum
17. Alıştırma ekranlarındaki geri bildirimler yeterliydi.
   - Kesinlikle katılmıyorum
   - Katılmıyorum
   - Fikrim yok
   - Katılıyorum
   - Kesinlikle katılıyorum
18. Alıştırma ekranlarındaki yönlendirmelerin motivasyonumu arttırmada etkiliydi.
   - Kesinlikle katılmıyorum
   - Katılmıyorum
   - Fikrim yok
   - Katılıyorum
   - Kesinlikle katılıyorum
19. Alıştırmalar, eğitimlerde öğrendiğim bilgiler ile alakalı değildi.
   - Kesinlikle katılmıyorum
   - Katılmıyorum
   - Fikrim yok
   - Katılıyorum
   - Kesinlikle katılıyorum
20. Alıştırma ekranlarında bazı ajanların hatalı cevaplarının konuyu öğrenmemde olumlu etkisi oldu.
   - Kesinlikle katılmıyorum
   - Katılmıyorum
   - Fikrim yok
   - Katılıyorum
   - Kesinlikle katılıyorum
21. Konuyu öğrenmemde, alıştırma ekranları eğitim ekranlarından daha çok yardımcı oldu.
   - Kesinlikle katılmıyorum
   - Katılmıyorum

- Fikrim yok
- Katılıyorum
- Kesinlikle katılıyorum

22. Sistem genel olarak UML konusundaki bilgimi arttırmada başarılıydı.
    - Kesinlikle katılmıyorum
    - Katılmıyorum
    - Fikrim yok
    - Katılıyorum
    - Kesinlikle katılıyorum

APPENDIX D

HCI LAB RESULTS

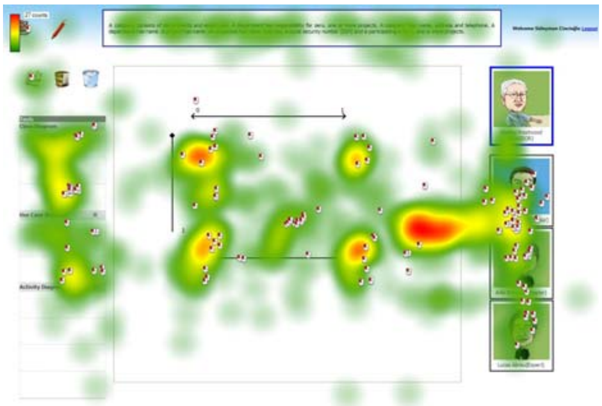

Figure 8.1 HCI participant 1 Class Diagram Practice

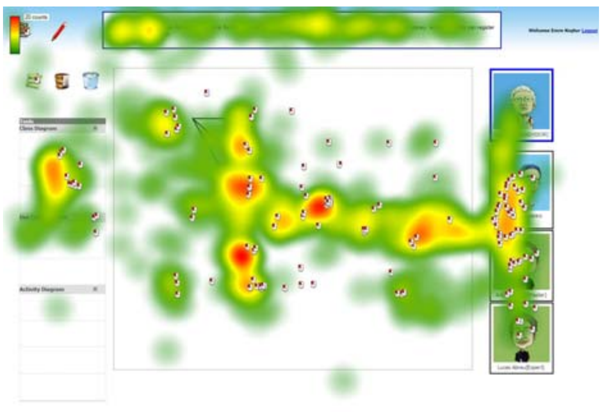

Figure 8.2 HCI participant 1 Use Case Diagram Practice

Figure 8.3 HCI participant 1 Activity Diagram Practice

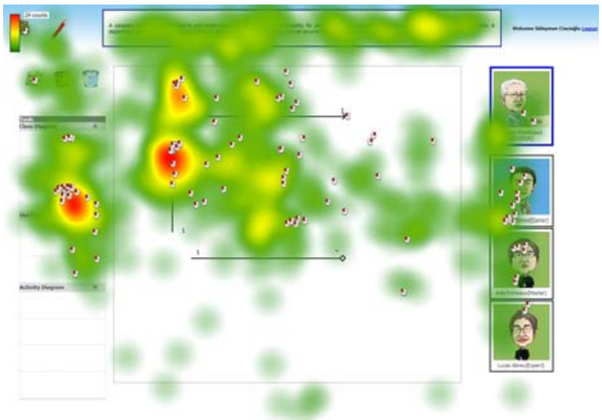

Figure 8.4 HCI participant 2 Class Diagram Practice

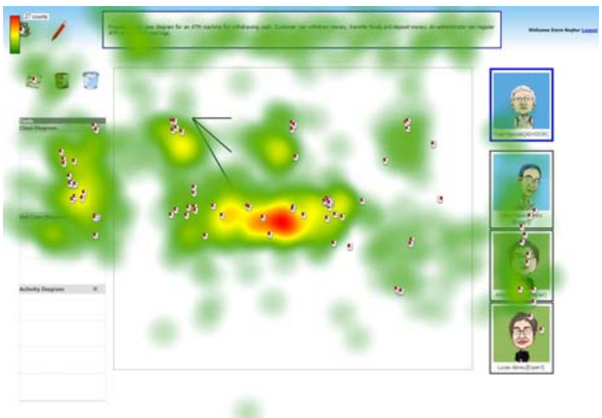

Figure 8.5 HCI participant 2 Use Case Diagram Practice

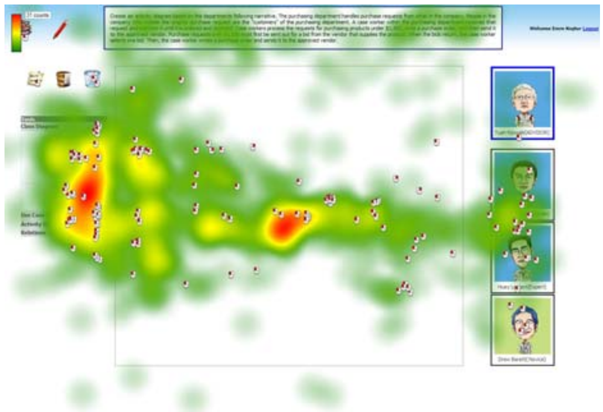

Figure 8.6 HCI participant 2 Activity Diagram Practice

Figure 8.7 HCI participant 3 Class Diagram Practice

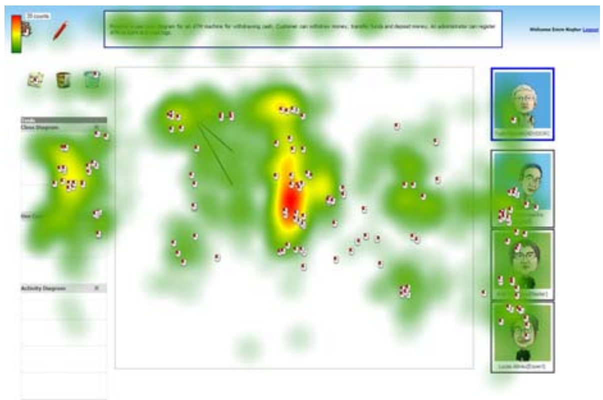

Figure 8.8 HCI participant 3 Use Case Diagram Practice

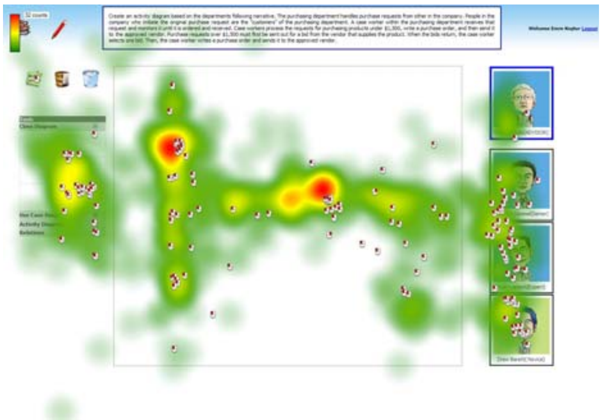

Figure 8.9 HCI participant 3 Activity Diagram Practice

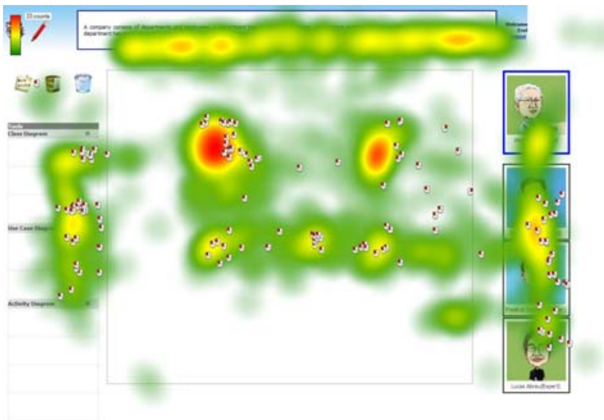

Figure 8.10 HCI participant 4 Class Diagram Practice

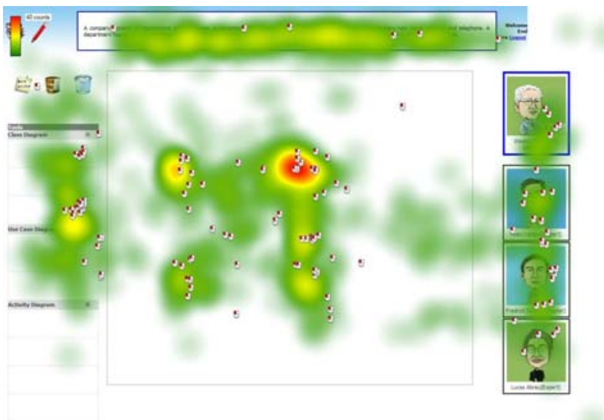

Figure 8.11 HCI participant 5 Class Diagram Practice

APPENDIX E

APPROVAL FROM ETHICS COMMITTEE

1956

Orta Doğu Teknik Üniversitesi
Middle East Technical University

Öğrenci İşleri Daire Başkanlığı
Registrar's Office

06531 Ankara, Türkiye
Phone: +90 (312) 2103417
Fax: +90 (312) 2107960
www.oidb.metu.edu.tr

B.30.2.ODT.72.00.00/420-3396-761 03/06/2010

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Üniversitemiz Bilgisayar Mühendisliği EABD Yüksek Lisans Programı öğrencisi Efe Cem Kocabaş'ın 17 Mayıs- 31 Mayıs 2010 tarihleri arasında *"UML ALF Agent Based Adaptive Learning Framework: A Case Study on ULM" başlıklı* araştırmasına ilişkin hazırlanan anketi, ODTÜ Bilgisayar Merkezi ve İnternet ortamında uygulama yapması için öğrencinin isteği doğrultusunda görevlendirilmesi Etik Komite onayı ile uygun görülmüştür.

Uygulamanın yapılabilmesi için gereğini arz ederim.

Saygılarımla.

Nesrin Ünsal
Öğrenci İşleri Daire Başkanı

EKLER:
1-Danışmanın dilekçesi
2-Öğrencinin dilekçesi
3-İAEK Başvuru Formu
4-Çalışmanın özeti
5-Anket

HA

ODTÜ
Fen Bil. Enst. Müdürlüğü
Evrakın GELİŞ TARİHİ 07 HAZ 2010
SIRA NO 1221

APPENDIX F

TEST QUESTIONS

CHAPTER 1 INTRODUCTION

1. Which is NOT true about UML?

- UML is a visual language for modeling and communicating about systems through the use of diagrams and supporting text.
- UML is a programming language.
- UML brings together the information systems and technology industry's best engineering practices.
- UML stands for Unified Modeling Language.

2. Which is NOT an aspect of UML?

- Language
- Methodology
- Unified
- Modeling

3. Which is a false statement about model?

- A model is a representation of a subject.
- A model captures a set of ideas known as abstractions about its subject.
- It provides a common understanding of the requirements and the system for team members.
- When creating a model, everything should be represented about the subject all at once.

4. The UML is a language for ... [2 answers]

- Specifying
- Manufacturing
- Constructing
- Programming

5. Which is not one of the methods of OMG's scope when creating the UML?

- Yourdon-Coad
- Booch Method
- Object-modeling technique (OMT)
- Object-oriented software engineering (OOSE)

6. Who are the 'Three Amigos'?

- Edward Yourdon - Grady Booch - Ward Cunningham
- James Rumbaugh - Grady Booch - Ivar Jacobson
- Ward Cunningham - Kent Beck - James Rumbaugh
- Ivar Jacobson - Kent Beck - Edward Yourdon

7. Which of the following lifecycle is in correct order?

- Analysis - Design - Deployment – Testing
- Design - Requirement Gathering - Implementation – Deployment
- Requirement Gathering - Analysis - Implementation – Testing
- Analysis - Design - Testing – Implementation

8. Which of the following statement is false?

- Within a waterfall approach, all the elements are captured and analyzed, and the whole system is designed, implemented, tested, and deployed in this linear sequence.
- The UML models must be fairly complete at each step.
- When applying an iterative approach, any subsets of the lifecycle activities are performed several times to better understand the requirements and gradually develop a more robust system.
- When applying an iterative approach, solving problems discovered during testing activities might be too late or too costly.

9. Which is NOT one of the benefits of an iterative approach?

- We can better manage complexity by building a system in smaller increments rather than all at once.
- We can better manage changing requirements by incorporating changes throughout the process and not trying to capture and address all the requirements at once.
- We can provide general solutions to users at the end of the process.
- We can solicit feedback from users concerning the parts of the system already developed, so we may make changes and guide our progress in providing a more robust system that meets their requirements.

10. Which of the following statement is false?

- A use case is a functional requirement described from the perspective of the users of a system.
- The elements and how they interact and collaborate is known as the system's structure. Modeling a system's structure is known as structural modeling.

- An architecture-centric process focuses on the architecture of a system across iterations.
- A risk is any obstacle or unknown that may hinder our success.

CHAPTER 2 OBJECT ORIENTED MODELING

1. Which of the following statements are true? [2 answers]

- A project uses requirements as an input and produces a system (or a part of a system) as an output.
- Database of the project management system executes on a client computer.
- When creating a project, start and end dates of the project is uncertain.
- Project becomes active when human resources are assigned to the project.

2. Which of the following statement is false?

- An alphabet defines the simplest parts of a language: letters, characters, signs, and marks.
- The smallest units of meaning in a language are its "letters".
- A language's syntax specifies the notation used for communication and is determined by the language's alphabet.

3. How can we simply describe the classes?

- The general concepts shown in the sentences
- The specific concepts
- The general relationships
- The specific relationships

4. Which of the facts below about the attributes IS NOT correct?

- Something that an object knows, which essentially represents data, is called an attribute.
- A class defines attributes and an object has values for those attributes.
- If two objects have the same values for their attributes, then they are counted as same object.
- Associations and attributes are known as structural features.

5. Which of the following facts about messages and stimuli IS NOT correct?

- A stimulus is an instance of a message similar to how an abject is an instance of a class.
- The sender is known as the client.
- The receiver is known as the supplier.
- Communication between objects via their links is called a message.

6. Encapsulation is also known as...

- Localization
- Information hiding
- Abstraction
- Generalization

7. The ability to have multiple methods for a single operation is called...
- Encapsulation
- Generalization
- Polymorphism
- Abstraction

8. Which of the following diagrams DOES NOT belong to Structural Modeling?
- Use-case diagrams
- Sequence diagrams
- Deployment diagrams
- Class diagrams

9. Which of the following IS one of the types of elements of Activity Diagrams?
- An event
- Associations
- A lifeline
- A swimlane

10. The UML diagrams may be generally organized around:
- The use-case or user architectural view
- The structural or static architectural view
- The behavioral or dynamic architectural view
- All of the above

CHAPTER 3 CLASS AND OBJECT DIAGRAMS

1. Which of the following statement is false?
- Class modeling is a specialized type of modeling concerned with the general structure of a system.
- Object modeling is a specialized type of modeling concerned with the structure of a system at a particular point in time.
- Object modeling is usually used in conjunction with class modeling to explore and refine class diagrams.
- Class and object modeling usually start before the requirement analysis.

2. Which of the following statements are true? [2 answers]
- Structural features define what objects of the class know.
- Behavioral features define what objects of the class can do.
- Structural features include operations and methods.
- Behavioral features include attributes and associations.

3. Which of the following statement is true?
- An attribute is what an object of a class can do. It is a specification of a service provided by the object.
- An operation is what an object of a class knows. It's an element of data maintained by the object.
- In a class diagram, attributes are listed in the second compartment for a class.
- In a class diagram, operations are listed in the first compartment for a class.

4. Which of the following fact about objects is false?
   - An object is a general concept. It defines a type of class and its characteristics.
   - An object has structural features including attribute values and links.
   - An object has behavioral features including operations and methods.
   - An object is shown as a solid-outline rectangle with two standard compartments separated by horizontal lines.

5. Which of the following fact about associations is false?
   - An association defines a type of link and is a general relationship between classes.
   - A binary association relates two classes.
   - A binary association may be labeled with a name.
   - A binary association should be named using a noun phrase.

6. Which of the following facts about multiplicity is true? [2 answers]
   - It indicates how many objects of a class may relate to the other classes in an association.
   - Default multiplicity for association end is zero or more.
   - Multiplicity is a must.
   - Multiplicity is simply undefined, unless it is specified.

7. Which of the following statement about aggregation is false?
   - Aggregation is whole-part relationship between an aggregate, the whole, and its parts.
   - Aggregation is often known as a has-a relationship.
   - Aggregation is shown using a filled-in diamond attached to the class that represents the whole.
   - Aggregation is used only with binary associations.

8. Which of the following statement about composition is false?
   - Composition is whole-part relationship between a composite and its parts.
   - Composition is often known as a has-a relationship.
   - Composition is shown using a filled-in diamond attached to the class that represents the whole.
   - Composition is used only with binary associations.

9. Which is not a rule for representing links?
   - Label links with their association names, and underline the names to show that they are specific instances of their respective associations.
   - Ensure that link ends are consistent with their corresponding association ends.
   - Translate association multiplicity into one or more specific links between specific objects.
   - A link relates two classes.

CHAPTER 4 USE CASE DIAGRAMS

1. Which of the following fact about use cases is false?
   - Given that every project has limited resources, you can use this information about use cases to determine how best to execute a project.

- Use cases cannot be reused.
- Use-case modeling is a specialized type of structural modeling concerned with modeling the functionality of a system.
- An understanding of how use cases are related allows users and analysts to negotiate and reach agreement concerning the scope and requirements of a system.

2. Which is not an element of a use case diagram?
   - Use Case
   - Activity
   - Actor
   - Extends

3. Which element is shown with the specified figure?
   - Use Case
   - Admin
   - Actor
   - User

4. Which of the following statements are true? [2 Answers]
   - An actor can be shown as a class marked with the actor keyword and labeled with the name of the actor class.
   - An actor may be a human user or another system.
   - An actor does not represent a resource owned by another project or purchased rather than built.
   - When speaking of a specific actor of a class, the term actor class is used.

5. Which of the following statement is false?
   - Use case is shown as an ellipse and labeled with the name of the use-case class.
   - Use cases may be enclosed by a rectangle that represents the boundary of the system that provides the functionality.
   - A use case is a functional requirement that is described from the perspective of the users of a system.
   - When speaking of a class of use cases, it's customary to use the term use-case instance.

6. What is communicate associations?
   - It is a functional requirement that is described from the perspective of the users of a system.
   - It indicates how many use-case classes may relate to the other classes in an association.
   - It addresses the question of how actors and use cases are related and which actors participate in or initiate use cases.
   - It is the performance of a specific sequence of actions and interactions.

7. Which of the following fact about include dependency is false?
   - An include dependency from one use case to another use case indicates that the base use case will include or call the inclusion use case.

- An include dependency is shown as a dashed arrow from the inclusion use case to the base use case marked with the include keyword.
- The base use case is responsible for identifying where in its behavior sequence or at which step to include the inclusion use case.
- An include dependency can be used when a use case may be common to multiple other use cases and is therefore factored out of the different use cases so that it may be reused.

8. Which of the following fact about extend dependency is false?
    - An extend dependency can be used to address the situation by factoring out optional behavior from a use case.
    - A use case may extend multiple use cases, and a use case may be extended by multiple use cases.
    - An extend dependency from one use case to another use case indicates that the extension use case will extend and augment the base use case.
    - An extend dependency is shown as a solid arrow from the extension use case to the base use case marked with the extend keyword.

CHAPTER 5 COMPONENT AND DEPLOYMENT DIAGRAMS

1. What is the difference between Component modeling and Deployment modeling?
    - Component modeling is structural, Deployment modeling is behavioral.
    - Deployment model shows the external resources that those components require.
    - Component modeling typically starts after the design of the system is fairly complete.
    - Deployment modeling is applied during design activities.

2. Which of the following can be a component of a system?
    - A user interface component.
    - A business-processing component.
    - A data component.
    - All of the above.

3. Which of the following facts about components is false?
    - A component is shown as a rectangle. The rectangle is labeled with the name of the component class fully underlined.
    - A component is shown as a rectangle with two small rectangles protruding from its side.
    - A component instance is shown similar to a component class, but is labeled with the component instance name followed by a colon followed by the component class name.
    - Both names are optional, and the colon is present only if the component class name is specified.

4. Which of the following facts about nodes is false?
    - A node is a resource that is available during execution time.
    - A node may be a computer, printer, server, Internet, or any other kind of resource available to components.

- A node is shown as an ellipse labeled with the node's name.
- When speaking of a specific component of a class, the term node instance is used.

5. Which of the following statement is false?
    - A reside dependency from a component to any UML element indicates that the component is a client of the element, which is itself considered a supplier, and that the element resides in the component.
    - A use dependency from a client component to a supplier component indicates that the client component uses or depends on the supplier component.
    - A use dependency from a client component to a supplier component's interface indicates that the client component uses or depends on the interface provided by the supplier component.
    - A deploy dependency from a client component to a supplier node indicates that the supplier component is deployed on the client node.

CHAPTER 6 SEQUENCE AND COLLABORATION DIAGRAMS

1. Which of the following fact about sequence diagrams is false?
    - A sequence diagram shows elements as they interact over time, showing an interaction or interaction instance.
    - Sequence diagrams are organized along two axes: the horizontal axis shows the elements that are involved in the interaction, and the vertical axis represents time proceeding down the page.
    - The elements on the vertical axis may appear in any order.
    - Sequence diagrams are made up of a number of elements, including class roles, specific objects, lifelines, and activations.

2. Which of the following elements are the elements of sequence diagram? [2 answers]
    - Class roles
    - Association roles
    - Lifelines
    - Specific links

3. Which of the following element is not an element of collaboration diagram?
    - Class roles
    - Specific objects
    - Specific links
    - Activation

4. Which element is shown with the specified figure?
    - Class roles
    - Specific objects
    - Lifelines
    - Activation

5. Which of the following statement is false?
    - Reflexive communication is an element that communicates with itself where a communication is sent from the element to itself.

- When an element is created during an interaction, the communication that creates the element is shown with its arrowhead to the element.
- When an element is destroyed during an interaction, the communication that destroys the element is sent back from the element to communication's lifeline where the destruction is marked with a large "X" symbol.
- Repetition (which involves repeating a set of messages or stimuli) within a generic-form interaction is shown as a set of communications enclosed in a rectangle.

6. Which is not a difference between collaboration and sequential diagram?
   - A collaboration diagram shows elements as they interact over time.
   - Collaboration diagrams are time and space-oriented and emphasize the overall interaction, the elements involved, and their relationships.
   - Sequence diagrams are useful for complex interactions.
   - Collaboration diagrams are useful for visualizing the impact of an interaction on the various elements.

CHAPTER 7 STATE DIAGRAMS

1. Which of the following statement is false?
   - Each element has a lifecycle.
   - A state is a specific condition or situation of an element during its lifecycle.
   - The current state of an element is called its active state.
   - A state diagram may have only one initial and final state.

2. Which of the following statement is false?
   - An initial state indicates the state of an element when it is created and it is shown using a small solid filled circle.
   - A final state indicates the state of an element when it is destroyed and it is shown using a circle surrounding a small solid empty circle.
   - There are various types of states, including simple, initial, and final states.
   - State modeling is a specialized type of behavioral modeling concerned with modeling the lifecycle of an element.

3. Which is not a simple state?
   - Inactive
   - Active
   - Suspended
   - Terminated

4. Which of the following facts about transitions is false?
   - A transition without an event and action is known as an automatic transition.
   - When an event occurs, the transition is said to fire.
   - A transition is shown as a dashed line from a source state to a target state labeled with the event followed by a forward slash followed by the action.
   - Transitions between states occur as follows: An element is in a source state. Then an event occurs. After that an action is performed. Finally, the element enters a target state.

5. Which of the following statement is false?
  - An event is an occurrence, including the reception of a request.
  - An action represents processing.
  - A state may be shown as a circle.
  - It is allowed to put one or more state diagrams inside a single state to indicate that when an element is in that state, other elements inside of it have their own states.

CHAPTER 8 ACTIVITY DIAGRAMS

1. Which of the following statement is false?
  - Each element has the responsibility of appropriately reacting to the communications it receives.
  - An action state represents processing as an element fulfills a responsibility.
  - A simple action state represents processing.
  - An activity diagram may have more than one initial and final state.

2. Which of the following statement is false?
  - An initial action state indicates the first action state on an activity diagram. It is shown using a small solid empty circle.
  - A final action state indicates the last action state on an activity diagram and it is shown using a circle surrounding a small solid filled circle (a bull's eye).
  - There are various types of action states, including simple, initial, and final action states.
  - Activity modeling is a specialized type of behavioral modeling concerned with modeling the activities and responsibilities of elements.

3. Which is not an activity diagram element?
  - Initial Action State
  - Final Action State
  - Class
  - Decision

4. Which of the following facts about transitions is false?
  - A flow transition shows how action states are ordered or sequenced.
  - A control-flow transition indicates the order of action states.
  - An object-flow transition indicates that an action state inputs or outputs an object.
  - Object-flow transitions are also known as default transitions or automatic transitions.

5. Which of the following facts about transitions is false?
  - A swimlane is a visual region in an activity diagram that indicates the element that has responsibility for action states within the region.
  - A decision involves selecting one control-flow transition out of many control-flow transitions based upon a condition.
  - Concurrency involves selecting multiple transitions simultaneously.
  - If a bar has one incoming transition and two or more outgoing transitions, it indicates that all outgoing transitions occur once the incoming transition occurs. This is called synchronization of control.