A MULTI-AGENT ADAPTIVE LEARNING SYSTEM FOR DISTANCE EDUCATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATMA CEMİLE SERÇE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

THE DEPARTMENT OF INFORMATION SYSTEMS

JANUARY 2008

Approval of the Graduate School of Informatics.

Prof. Dr. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Yasemin Yardımcı
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Ferda Nur Alpaslan
Supervisor

Examining Committee Members

Prof. Dr. Nazife Baykal

Assoc. Dr. Ferda Nur Alpaslan

Assoc. Prof. Dr. Soner Yıldırım

Assist. Prof. Dr. Yasemin Gülbahar

Dr. Altan Koçyiğit

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 wok.

Name, Lastname: Fatma Cemile Serçe

Sign:

ABSTRACT

A MULTI-AGENT ADAPTIVE LEARNING SYSTEM FOR
DISTANCE EDUCATION

Serçe, Fatma Cemile
Ph.D., Department of Information Systems
Supervisor: Assoc. Prof. Dr. Ferda Nur Alpaslan

January 2008, 124 pages

The adaptiveness provides uniquely identifying and monitoring the learner's learning activities according to his/her respective profile. The adaptive intelligent learning management systems (AILMS) help a wide range of students to achieve their learning goals effectively by delivering knowledge in an adaptive or individualized style through online learning settings. This study presents a multi-agent system, called MODA, developed to provide adaptiveness in learning management systems (LMS). A conceptual framework for adaptive learning systems is proposed for this purpose. The framework is based on the idea that adaptiveness is the best matching between the learner profile and the course content profile. The learning styles of learners and the content type of learning material are used to match the learner to the most suitable content.

The thesis covers the pedagogical framework applied in MODA, the technical and multi-agent architectures of MODA, the TCP-IP based protocol providing communication between MODA and LMS, and a sample application of the system to an open source learning management system, OLAT. The study also discusses the possibilities of future interests.

Keywords: Adaptive Learning Systems, Intelligent Learning Management Systems, Multi-agent Systems, Distance Learning, Fusion of Agents and Learning.

ÖZ

UZAKTAN ÖĞRENMEYE YÖNELİK ADAPTE OLABİLEN ÇOK ARACILI SİSTEM

Serçe, Fatma Cemile
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Assoc. Prof. Dr. Ferda Nur Alpaslan

Ocak 2008, 124 sayfa

Adapte olabilen akıllı öğrenme yönetim sistemleri, farklı bilgi birikimine sahip çok sayıda öğrenciye, uzaktan öğrenme ortamları aracılığıyla adapte olabilen etkin ve akıllı öğrenme ortamı sunmaktadır. Bu sistemler, ders içerikleri ve diğer öğrenme kaynakları arasında öğrencinin profiline en uygun olan içerikleri seçer ve öğrenciye sunar. Öğrencinin hareketleri sürekli takip edilir, davranışları sınıflandırılır ve profilleri güncelleştirilir. Bu çalışma da, öğrenme yönetim sistemlerine (ÖYS) bu adapte olabilme yeteneğini katacak MODA adında bir yazılım birimi geliştirilmiştir. MODA herhangi bir ÖYS tarafından kullanılabilecektir.

Bu çalısma kapsamında, uzaktan ögrenme ortamlarında kullanılacak adapte olabilen sistemlerin eğitimsel temelini oluşturacak bir kuramsal ve denenceli çerçeve tanımlanmıştır. Bu çerçeve, öğrencinin ve ders içeriğinin nasıl modellenmesi ve güncellenmesi gerektiği bilgilerini içerir. Öğrenciye en uygun içerik, öğrenci profiline en yakın ders içerik profillerinin hesaplanması ile bulunur. MODA sistemi, bu çerçeveyi temel almaktadır.

Bu çalışmada, MODA sisteminin teknik alt yapısı, aracılar, aracıların rolleri, aracılar arasındaki etkileşim, aracılar arasında iletişimi sağlayan ontoloji tanımlamaları ve her hangi bir ÖYS tarafından kullanılabilmeyi sağlayan ve yine bu çalısma kapsamında geliştirilen TCP-IP tabanlı iletişim kuralı detaylı olarak anlatılmıştır. MODA'nın bir ÖYS ile birleştirimi gerekleştirilmiştir. Bu çalışmanın sonunda, bu örnek birleştirimden edinilen çıkarımlar ve ileride yapılacak çalışmalara yönelik öneriler sunulmuştur.

Anahtar Kelimeler: Adapte Olabilen Öğrenme Sistemleri, Akıllı Öğrenme Yönetim Sistemleri, Çok Aracılı Sistemler, Uzaktan Öğrenme, Aracılar ve Öğrenme

To my mother, Ayten Hoşver …

ACKNOWLEDGMENTS

I am deeply indebted to my supervisor Assoc. Prof. Dr.Ferda Nur Alpaslan whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis.

I would like to thank the thesis committee members, Prof.Dr. Nazife Baykal, Dr.Ali Arifoğlu, Assoc. Prof. Dr. Soner Yıldırım and Assist.Prof.Dr. Yasemin Gülbahar for their comments and suggestions.

I would like to express my gratitude to Professor İbrahim Akman, who gave me the opportunity to work with them in the Department of Computer Engineering at Atılım University and to go ahead with my thesis.

I wish to thank Assist.Prof.Dr. Nergiz Çağıltay, Assist.Prof.Dr. Ciğdem Turhan, Fügen Selbes and Gül Tokdemir. They helped me immensely by giving me encouragement and friendship.

I express my deepest appreciation to my husband Hakan Serçe for his help, support, interest and valuable contributions. His patient love enabled me to complete this work. One of the best thing that we lived through in PhD period was the birth of our beautiful girly Ebru, who has brought so much happiness and joy into our life.

I owe my most sincere gratitude to my mother Ayten Hoşver, Dursun Ali Hoşver, Halil İbrahim Hoşver and Zehra Elif Alemdaroğlu gave me the possibility to complete this thesis. She supported and motivated me with her never ending patience, tolerance and understanding.

I would like to thank Mehmet Serçe and Miyase Serçe for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Background of the Study

Adaptiveness is a crucial issue in today's online learning environments (OLE). In[1], it is argued that virtual learning environments (VLE) are best at achieving learning effectiveness when they adapt to the needs of individual learners. In service-job-training especially necessitates to identify learning needs and customize solutions that foster successful learning and performance, with or without an instructor to supplement instruction. The learning management systems provide educational services to a wide range of students and they can help students to achieve their learning goals by delivering knowledge in an adaptive or individualized way[1]. In [2], it is stated that as long as the completion on the market of Web-based educational system increases, "begin adaptive" will become an important factor for winning the customers.

Web-based adaptive and intelligent education systems inherit many properties from Intelligent Tutoring Systems (ITS) and adaptive hypermedia systems[2]. The adaptive intelligent learning systems have the advantages of both artificial intelligence and learning management systems.

There are different adaptive technologies. Brusilovsky[2] provides some examples for adaptive systems such as InterBook[3], CALAT[4], ACE[5], ELM-ART II[6], ILESA[7] etc. He gives a detailed review of these systems with a focus on their supporting adaptive technologies are provided. The adaptive technologies includes ITS including curriculum sequencing, intelligent analysis of student's solutions, interactive problem solving support; adaptive hypermedia technologies involving adaptive presentation and adaptive navigation support; web-inspired technologies like student model matching.

There are some adaptive systems based on learning modeling and learning style, such as iWeaver[8], INSPIRE[9], MANIC[10], ARTHUR[11], CS388[12], AEC-ES[13], etc. ARTHUR, iWeaver, MANIC, CS388 base on sensory preferences; AEC-ES categorizes learners as either field-dependent (FD) or independent (FI) learners; CS388 functions in global-sequential dimension of Felder-Silverman[14]; and INSPIRE is based on Honey and Mumford model[15].

Pedagogical agents are autonomous agents that support human learning by interacting with students in the context of the learning environment. They extend and improve upon previous work on intelligent tutoring systems in a number of ways. They adapt their behavior to the dynamic state of learning environment, taking advantage of learning opportunities as they arise. They can support collaborative learning as well as individualized learning, because multiple students and agents can interact in a shared environment. The use of agents in providing adaptiveness has been experiences in some studies such as ADELE[16], PPP Persona[17], etc.

1.2 Purpose of the Study

E-learning becomes an industry and a research stream. The customers of e-learning industry are the learners having different background and personalities. The ultimate goal of e-learning is to improve the learner's learning and performance levels during online learning. To reach this goal, there is need for studies not only to develop infrastructures and to deliver information online, but also to improve learner's learning and performance. Adaptive learning system is a type of e-learning technology which behaves intelligently to understand the learner's needs and characteristics before delivering the content for them.

The purpose of this study is to develop a multi-agent system which provides adaptiveness for learning management systems. The secondary purpose is to construct a conceptual framework defining learner modeling, content modeling and adaptation strategies.

1.3 Significance of the Study

The intelligence of a learning system is largely attributed to its ability to adapt to specific student needs during the learning process. Online learning environments have been used by a much wider variety of students. Each student may have different backgrounds, learning styles, individual preferences, and knowledge levels. This raises the need for adaptiveness of learning environments. The learning systems must be flexible to be be suitable for any particular kind of students.

One of the basic requirements of education in the twenty-first century is to prepare students for participation in a knowledge-based economy. Knowledge is the most critical resource for social and economic development. It is the intellectual capital. So, the knowledge of a particular student grow very quickly. The content which is quite complex for a student at the beginning, may soon become quite trivial and boring to the same student. Therefore, the online learning environments is required to respond students according to their knowledge levels and to provide different form of knowledge presentation for different knowledge. The learning materials are needed to be presented according to the students preference and knowledge level, which makes the study processes more interesting and effective.

Teaching at a distance involves the use of different skills for the instructors comparing to the ones used in a traditional classroom. There is a consensus in the literature on the issue that getting the people to participate and making learning active at a distance is much more important than presenting the information. Online learning environments puts much work on the instructors shoulders. The instructors become less focused on the learning process. Because, the management and administration of learning setting takes too much time. Some examples for faculty barriers to distance learning are summarized as follows:

- the lack of support by the faculty. Faculty roles must change the most in administering distance learning programs. Faculty need to
  - change teaching styles to that of a mentor, tutor, and facilitator when necessary,
  - meet the needs of distance students without face-to-face contact,
  - change the course content to accommodate diverse student needs and expectations.
- lack the basic skills or hardware to fully participate in distance education,
- the difficulty in giving immediate feedback,
- the difficulty in assessing student performance.

Moreover, there is a need for providing education to students in different places around the world where no teacher is available for face-to-face assistance in most cases. To support teaching and to facilitate learning, learning management systems must provide individualized help just as a human tutor would.

The responsibilities of faculty and students in distance learning environments are more than in the traditional learning environments. Until recently, a major requirement of the existing learning management systems was to support both instructor and learners and to be ease of use. However, this seems to be no longer the main concern any more. There is a need for smart learning environments offering personal services with capabilities to learn, reason, have autonomy, and be totally dynamic. The adaptive technologies might diminish the barriers to learning in distance education. This study is an attempt to design and develop a learning system that enrich online learning management systems with adaptive instruction using agent technology.

1.4 Approach of the Study

All adaptive systems lie on a conceptual framework defining the adaptation mechanism. In this study, a conceptual framework for adaptive learning management systems was defined. This framework describes the ways to model the learner and content, and also to find the best match between the learner and content. We model the learner according to three factors which are behavioral factors, knowledge factors, and personality factors. We modeled the course content using 30 content types. These content types are derived from previous studies, IEEE LOM Metadata[18] information, and the descriptions of Felder-Silverman[14] learning style model. We conducted a study to classify the content with the learning style and we defined the learning resources and the corresponding learning style dimensions. The adaptation strategy in the framework is to find the best match between the learner and the instruction set. Using the classification information, we find the best match between the learner profile and the course profile by applying a normalized Euclidian distance function.

Based on this framework, we designed and developed an adaptive multi-agent system. There are different agents each having specific roles in promoting adaptive behavior. JADE[19] framework was chosen to implement these software agents. (For more information, one can refer to http://jade.tilab.com.). Gaia[20] was chosen as an agent-oriented software engineering technique to analyze and design the multi-agent system.

In literature, there are many adaptive systems providing powerful adaptive features. However, it is not possible to integrate most of these systems with the existing learning management systems. Most of them were developed to function as stand alone systems. There are lots of learning management systems used in practice and it might be very effective to plug adaptive features to these already existing and widely used learning management systems. In this research, the adaptive learning system was designed to be used with any learning management system. To achieve this interoperability, a communication protocol was defined to establish communication between the learning management systems and the adaptive multi-agent system, namely MODA. The system was integrated into an open source learning management system, called OLAT[21]. It is possible to obtain more information about OLAT at http://www.olat.org/website/en/html/index.html.

1.5 Road Map

Chapter 2 provides the review of literature about adaptive learning systems. Chapter 3 presents the basic concepts about the agent technology, and a review of pedagogical agents. These two chapters presents the detailed description of the previous studies on adaptive systems and agents which constitutes the background of the current thesis.

Chapter 4 explains the conceptual framework developed in this study. The framework covers the learner modeling, content modeling, and the learner-content matching strategies. The framework also provides the mechanism to initialize and update the learner and content profiles.

Chapter 5 contains the implementation details of the adaptive learning system, MODA. The system's agents, the interactions among the agents, the ontologies developed to establish agent communications are explained in this chapter. This chapter focuses mostly on the technical architecture of the system. This chapter also provides screen shots of integration of the system into the open source learning management system, called OLAT.

Chapter 6 provides the discussion for the conclusion and possibilities for the further research.

CHAPTER 2

Adaptive Instructional Systems

2.1 Learning and Instruction

Learning is a general term used for a lasting change in our behavior caused by an experience(Gagne, 1985 in [22]). It is the development of new skills, knowledge, or attitudes as we interact with the information and environment. Learning takes place when a lasting change of behavior takes place. If the instruction is not new to us, we have been previously engaged in this material and have already learned the material. Therefore, we are not learning something new but possibly revisiting the old information. It also must be a lasting change of behavior so we can apply and use this information on demand such as completing an assignment or taking a test[22].

Different educational psychologies view the concept of learning differently. Behaviorists believe that learning is nothing but change in behavior; cognitive theorists view learning as a process; and social learning theorists view learning process as interaction/observation in social context (Meria and Caffarella, 1991 in [23]). One thing they have in common is that they all assume that "instruction will bring about learning" and, based on this assumption, instructional designers use theories as guidance to design effective instruction to bring about maximum learning (Driscol, 2002 cited in [23]).

Bloom defines three domains in which learning occurs: the cognitive, psychomotor and affective domains. According to Bloom, six types of learning are in the cognitive domain, each one building on the previous one. These include knowledge, comprehension, application, analysis, synthesis and evaluation[Bloom, 1956 cited in [23]].

Learning is relatively permanent change in behavior due to experience. Instruction provides conditions for learning, it never provides learning. Learning is an internal process performed by students. On the other hand, instruction is an external phenomenon. Instructional design is a deliberate process that tries to control and direct learning toward predictable ends. The designer tries to overcome the learning deficiency and to produce a plan specifying the instructional events and materials that will provide the conditions for learning. What an instructional designer is able to do is limited to the choice and arrangement of external conditions that will help the internal process of learning to occur[23].

Web-based instruction is a hypermedia-based instructional program. It uses the resources of the World Wide Web to provide instruction. The WWW permits to use a computer to design and deliver instruction, using text, sound, data, video, etc. Distance education is the process of providing broad curricula using the WWW[23].

2.2 Learning Styles

In order to adapt the instruction based on the learner's needs, we need to understand the learner. If we know the learner well, then it might be much easier to find the ways to meet the needs of the learner.

Learners have different ways of perception, construction and retention of knowledge[24]. Each learner has a unique learning process, because each has different prior knowledge, mental abilities, and personality factors.

Individuals perceive and process knowledge in different ways. This leads to the theory defined as "Learning Styles Theory". The learning styles theory begun with Jung[24] who underlined the major differences between individuals in terms of perception, decision and interaction. [25] have also followed this study and focused on understanding the differences in learning.

According to the learning styles theory, each learner has different ways of perception, and one style does not address all individuals. Therefore, instruction must be presented in different ways according to these differences. In other words, instructors must ask how can this learner achieve more? instead of why is this learner not a high-achiever?[26].

In [27], the author provides a definition table of similar terms relating to learning styles:

Table 2.1: Definitions of similar terms relating to learning styles

Term	Explanation
Learning Preference	favouring one method of teaching over another
Learning Strategy	adopting a plan action in the acquisition of knowledge, skills or attitudes
Learning Style	adopting a habitual and distinct mode of acquiring knowledge
Cognitive strategy	adopting a plan of action in the process of organising and processing information
Cognitive style	a systematic and habitual mode of organising and processing information

In literature, there are mainly five different learning style models referenced in the studies of the other adaptive learning systems. These are as follows:

- The Myers-Briggs Type Indicator[25]

- Kolb's Learning Style Model[28]
- Honey and Mumford's Typology of Learners[15]
- Felder-Silverman Model[14]
- Dunn, Dunn and Price Model[29]

2.2.1 The Myers-Briggs Type Indicator

The essence of the theory is that much seemingly random variation in the behavior is actually quite orderly and consistent, being due to basic difference in the way individual prefer to use their perception and judgment [30].

The personality types provided by this style is summarized in Table 2.2.

2.2.2 Kolb's Learning Style Model

Kolb's learning style model is grounded on John Dewey's experiential learning theory, Kurt Lewin's work stressing the importance of being active in learning, and Jean Piaget's theory on cognitive development as the result of the transaction between people and their environment (e.g. education, career, job role)[28].

Kolb's Learning Style model classifies learners as active (learning through concrete experience), reflective(learning through reflective observation), experimental(learning through active experimentation) and, theorizing(learning through abstract conceptualization).

The characteristics of each learner type is listed below.

Active Learners, *I want to get on and do things...*

- learn by trial and error
- tend to be impatient and want to do things for themselves rather than wait and be told how to do them

Table 2.2: The Myers-Briggs Type Indicator [25]

focusing attention	**(E)XTRAVERSION** People who prefer Extraversion tend to focus on the outer world of people and things	**(I)NTROVERSION** People who prefer Introversion tend to focus on the inner world of ideas and impressions
gathering information	**(S)ENSING** People who prefer Sensing tend to focus on the present from their senses	**(I)NTUITION** People who prefer Intuition and on concrete information gained tend to focus on the future, with a view toward patterns and possibilities
making decisions	**(T)HINKING** People who prefer Thinking tend to base their decisions primarily on logic and on objective analysis of cause and effect	**(F)EELING** People who prefer Feeling tend to base their decisions primarily on values and on subjective evaluation of person-centered concerns.
dealing with the outer world	**(J)UDGING** People who prefer Judging tend to like a planned and organized approach to life and prefer to have things	**(P)ERCEIVING** People who prefer Perceiving tend to like a flexible and spontaneous approach to life and prefer to keep their options open.

- gives spontaneous answers
- quality move to something new
- slow, methodological works bores them
- take the lead to push ahead

Reflective Learners, ***I want to think about things...***

- adopt a "wait an see" approach
- trying to think things through and do not give the first answer they come across but require more information
- tend to be uncertain about what to do
- confer wiht other people to see what their opinions are

Experiential Learners, ***I want to see if there is not a better way of doing things...***

- seek to find new ways of doing thing
- even if they like to be shown how to do something, they need to put their newly acquired knowledge immediately into practice
- what is important to them is finding the most effective way of putting into practice what they know
- tend to be energetic, impatient
- do not hesitate to take short cuts in solving problems
- new challenge are seen as new possibilities for learning

Theorizing Learners, ***I want to understand things...***

- try to build an all encompassing logical system

- question assumptions and make rules from different cases in thinking problems through step by step
- "concrete" examples are perceived as being too limited to understand the general whole
- their effort goes into making coherent pictures of complex situations
- try to detach themselves from emotions and personal opinions
- less sympathetic to the feelings of others

2.2.3 Honey and Mumford's Typology of Learners

Honey and Mumford (1992) created their own version of Kolb's classification - activists, practical, theorists and reflectors - after revisiting the work of Kolb[15].

The Honey and Mumford's original definitions of the classification and some important consequence from practionners' websites (e.g. [31] [15] [32]) are as follows:

Activists The activists involve themselves fully and without bias in new experiences. Their philosophy is: I'll try anything once. They are willing to work with others but, want themselves to be in the center all activities. They learn best when they can immediately do something. They learn least when the they have to listen to long explanations, absorb a lot of data, etc. They like pedagogical activities such as: brainstorms, problem solving, group discussions, role plays, competitions, etc.

Reflectors Reflectors like to observe tasks from many different perspectives. They collect data, both first hand and from others, and prefer to think about it thoroughly before coming to a conclusion. They learn best when they can observe, review and think about what is happening. They learner least when they have to act as leaders. They like asynchronous discussion, observing activities, paired discussions, coached activities, questionnaires and interviews etc.

Theorists This type of learners adapt and integrate observations into complex but logically sound theories. They think problems through in a vertical, step-by-step logical way. They are logical in their learning. They learn best when they can study theories, models, concepts, stories etc. They learn least when the activity is ill structured, no principles are taught. They like discussions on theories, background information etc.

Pragmatists Pragmatists are keen on trying out ideas, theories and techniques to see if they work in practice. They positively search out new ideas and take the first opportunity to experiment with applications. Their philosophy is "There is always a better way" and "If it works it's good". They learn best when they use new information in real-life problems. They learn least from the theories. They like case studies, discussions and problem solving activities.

2.2.4 Felder-Silverman Model

Felder and Silverman model provides eight learning styles which are described as follows[14]:

Active Active learners like to try things out and see how they work and like to work with others.

Reflective Reflective learners like to think things through first.

Sensing Sensors like to learn facts, use well established methods practically and carefully.

Intuitive Intuitors tend to work fast and be innovative and can often handle abstract and mathematical concepts well.

Visual Visual learners like diagrams, pictures, graphs and films.

Verbal Verbal learners get more out of words heard and written.

Sequential Sequential learners like to work in linear steps that follow logically.

Global Global learners like to jump in, absorb material nearly at random and then get the big picture.

The Felder and Silverman learning style model categorizes a student's learning style on a sliding scale of four dimensions. These are:

- Active-Reflective
- Sensing-Intuitive
- Visual-Verbal
- Sequential-Global

Felder and Silverman also provide teaching style model, classifying instructional methods according to how well they address the proposed learning style components. The Figure 2.1 summarizes the dimensions of learning and teaching styles.

In this study, this learning style was taken as the baseline in modeling the learner. This learning style learning style model is accompanied by Felder-Solomon Index of Learning Style (ILS) instrument which to categorize individual learning style preferences. The questionnaire used consists of 44 questions, 11 questions for each of four dimensions. The advantages of this model are as follows:[33]:

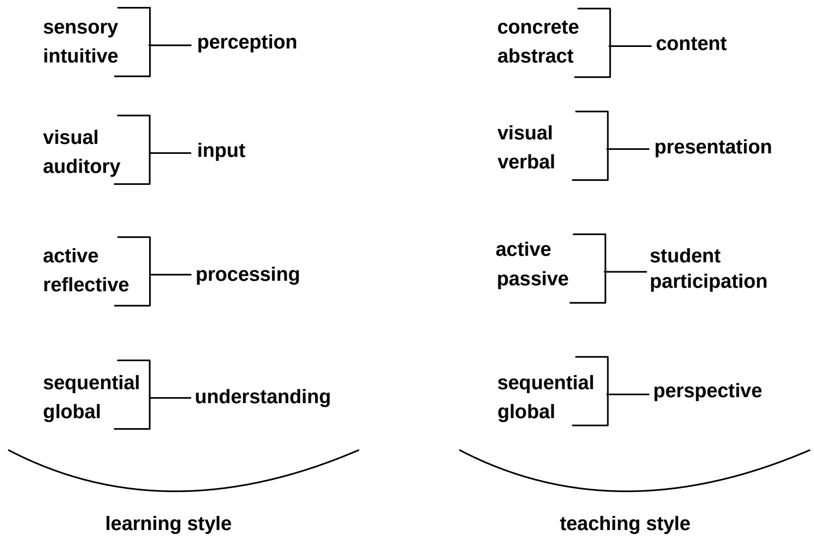


Figure 2.1: Dimensions of Learning and Teaching Styles[14]

1. The ILS has been validated[34].

2. The ILS questionnaire[35]provides a convenient and practical approach to establish the preferred learning style of each student. It is simple, easy to use and the results are easy to interpret.

3. The number of dimensions of the model is constrained, improving the feasibility of its implementation.

4. The results of ILS can be linked easily to adaptive environments(Paredes and Rodriguez, 2002 in [33]).

5. It is most appropriate and feasible to be implemented for hypermedia courseware(Carver and Howard, 1999 in [33]).

2.2.5 The Dunn and Dunn Learning Style Model

Dunn[29] described learning style as the way each learner begins to concentrate, process, and retain new and difficult information. Dunn and Dunn suggested that productivity style theorizes that each individual has a biological and developmental set of learning characteristics that are unique.

They further suggested that improvements in productivity and learning will come when instruction is provided in a manner that capitalizes on an individuals learning strengths.

The Dunn and Dunn Learning Style Model is based on five different categories:

1. Environmental
2. Emotional
3. Sociological
4. Physiological
5. Psychological

2.3 Adaptive Learning and Adaptive Instruction

Adaptive learning enables learners to select their modular components to customize their learner-centric learning environments. It offers flexible solutions that dynamically adapt content to fit individual's real-time learning needs[36].

Instructional approaches and techniques promoting adaptive learning are called adaptive instruction[36]. In the literature, the term adaptive instruction has been interchangeably used with individualized instruction. However, they are different concepts. Any type of instruction presented in a one-on-one setting can be considered as individualized instruction, but if it is not flexible enough to meet the student's specific learning needs, it can not be considered as adaptive.

In [37], the authors mention three essential ingredients of adaptive instruction, namely, providing a variety of alternatives for learning and many goals from which to choose, attempting to utilize and develop capabilities that an individual brings to the alternatives for his or her learning and to adjust to the learners particular talents, strengths, and weaknesses, and attempting to strengthen an individuals ability to meet the demands of available educational opportunities and develop skills necessary for success in the complex world.

2.3.1 History of Adaptive Learning Systems

Web-based Adaptive and Intelligent Educational Systems (AIES) inherit their characteristics from two earlier kinds of two AIES: intelligent tutoring systems (ITS) and adaptive hypermedia systems[2].

2.3.1.1 Intelligent Tutoring Systems

Intelligent tutoring systems (ITS) are adaptive instructional systems developed with the application of Artificial Intelligence(AI) methods and techniques. ITS provides learner oriented design and much more pedagogical knowledge implemented in the system[38]. The benefits of individualized instruction are the essence of ITS, which uses artificial intelligence to tailor multimedia learning.

In order to provide hints, guidance, and instructional feedback to learners, ITS systems typically rely on three types of knowledge, organized into separate software modules (see Figure 2.2). The *expert model* represents subject matter expertise and provides the ITS with knowledge of what it's teaching. The *student model* represents what the user does and doesn't know, and what he or she does and doesn't have. This knowledge lets the ITS know who it's teaching. The *instructor model* enables the ITS to know how to teach, by encoding instructional strategies used via the tutoring system user interface[38].

An **expert model** is a computer representation of a domain expert's subject matter knowledge and problem-solving ability. This knowledge

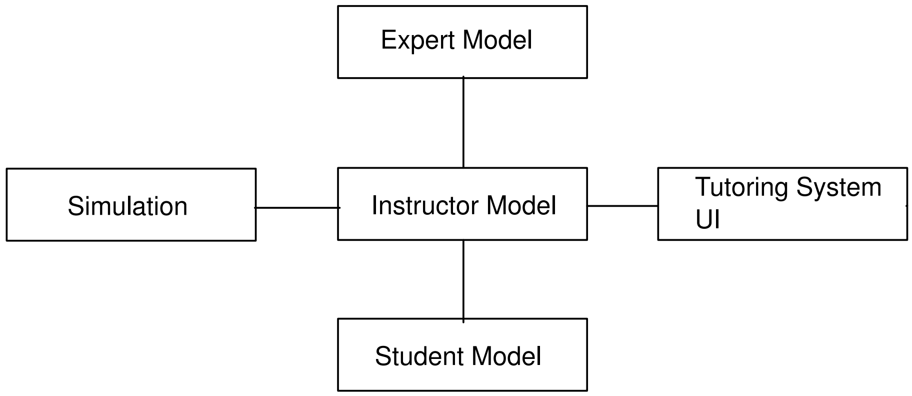


Figure 2.2: Components of an Intelligent Tutoring System[38]

enables the ITS to compare the learner's actions and selections with those of an expert in order to evaluate what the user does and doesn't know.

The **student model** evaluates each learner's performance to determine his or her knowledge, perceptual abilities, and reasoning skills. In more complex domains, the tutoring system can monitor a learner's sequence of actions to infer his or her understanding.

The **instructor model** encodes instructional methods that are appropriate for the target domain and the learner. Based on its knowledge of a person's skill strengths and weaknesses, participant expertise levels, and student learning styles, the instructor model selects the most appropriate instructional intervention. For example, if a student has been assessed a beginner in a particular procedure, the instructor module might show some step-by-step demonstrations of the procedure before asking the user to perform the procedure on his or her own. It may also provide feedback, explanations, and coaching as the participant performs the simulated procedure. As a learner gains expertise, the instructor model may decide to present increasingly complex scenarios. It may also decide to take a back seat and let the person explore the simulation freely, intervening with explanations and coaching only upon request. Additionally, the instructor model may also choose topics, simulations, and examples that address the user's competence gaps.

[39] propose an intelligent agent to guide students throughout the course material in the Internet. They designed an adaptive hypermedia system. The system functions as a personal assistant to help teachers to generate course curriculum and to help students to navigate through the course material. The teacher generates curriculum through a user interface, the curriculum is represented by a conceptual network and kept in a database table. Then, the concepts in a course are presented to the student via an intelligent user interface having the curriculum generated. The student is presented with a new concept if s(he) completed the concepts that re the prerequisites. The system constructs models of individual users, reflecting the user's state of knowledge. The system also has pretests at the end of each chapter. Analyzing the answers s(he) gave to the pretest, the student may be advised to restudy the concepts s(he) misunderstood or forgot. The study also conduct an experimental study on the system developed with the students of an undergraduate course offered in Computer Engineering Department at METU. According to the results, it was found that the system satisfies its objective to help the students to learn the course concepts efficiently with the adaptive navigational support.

Since those early implementations, ITSs have been developed for a widening variety of training applications. [38] summarizes some of these examples. An ITS developed by Alan M. Lesgold and colleagues at the University of Pittsburgh on behalf of a multinational semiconductor firm trains technicians to repair complex semiconductor chip-manufacturing equipment. Stottler Henke Associates, Incorporated (SHAI) developed an ITS for U.S. Navy officer tactical training using simulation and automated evaluation of each student's actions. Intelligent Tutoring Systems were announced and avowed as the future of education and training. Unfortunately, despite of the success of some ITSs such as Shutes Smithtown and Air Force, Clanceys GIDEON and NEOMYCIN, Wollfs Meno Tutor, Andersons LISP tutor, Nwanas FITS, Wagners SCHOLAR, Johansons PROUST, etc., ITSs have not yet seen general acceptance. [40] says that 10 years later, the ITSs community is still talking about the promise of this technology while searching for the leverage that will encourage its widespread adoption and classroom use. Much has to do with the complexities involved in the definition and design of ITSs applications, as well as the paradigmatic changes required of training and education organizations in the way they practice instructional design in order to realize this new kind of education paradigm [Clancey, 1996 cited in [40]].

2.3.1.2 Adaptive Hypermedia Systems

In [41], Brusilovsky defines adaptive hypermedia as an alternative to the traditional one-size-fits-all approach. In adaptive hypermedia systems, there is a model of the goals, preferences and knowledge of each individual user, which is used during the user's interaction with the system so that it adapts the hypertext to the meet the needs of the user.

Brusilovsky stated that the adaptive educational hypermedia was inspired by the area of intelligent tutoring systems and was designed to combine the advantages of an intelligent tutoring system (ITS) and an educational hypermedia[41]. This combination of ITS and educational hypermedia constitutes the adaptive hypermedia systems.

There are many adaptive hypermedia systems such as: ELM-ART[6], InterBook[3], ACE[5], Arthur[11], etc.

2.3.2 Adaptation Techniques

There are different adaptation techniques.Brusilovsky[42] explains the adaptive techniques under two main categories: adaptive hypermedia technologies and intelligent tutoring techniques (See Figure 2.3).

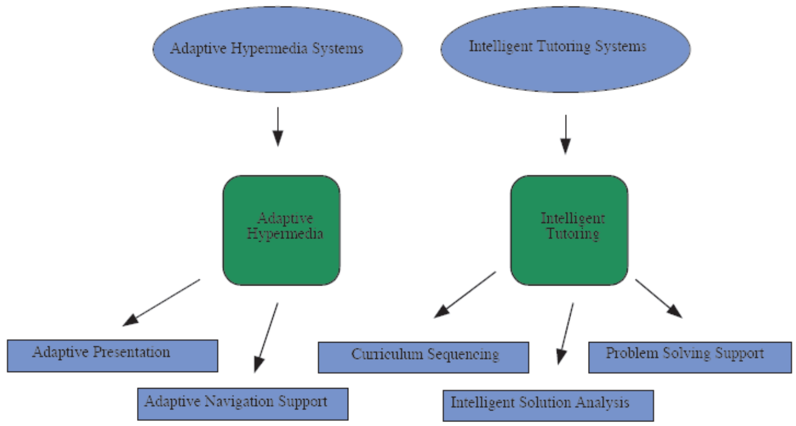


Figure 2.3: Adaptive Technologies[2]

The descriptions of the techniques in both adaptive hypermedia and ITS are as follows:

Adaptive Presentation adapting the content of a page accessed by a particular user to current knowledge, goals, and other characteristics of the user.

Adaptive Navigation Support assisting the learner in hyperspace orientation and navigation by changing the appearance of visible links.

- **Direct Guidance** deciding what is the next best node for the user to visit according users goal and other parameters represented in the user model.

- **Adaptive Sorting of Links** sorting all the links of a particular page according to the user model and to some user-valuable criteria.

- **Adaptive Hiding of Links** restricting the navigation space by hiding links to "not relevant" pages.

- **Adaptive Annotation of Links** augmenting the links with some form of comments which can tell the user more about the current state of the nodes behind the annotated links.

Curriculum Sequencing providing the learner with the most suitable individually planned sequence of contents to learn and learning tasks (examples, questions, problems, etc.) to work with.

Intelligent Solution Analysis considering learner's solutions of educational problems and telling what is wrong or incomplete and which missing or incorrect pieces of knowledge may be responsible for the error.

Problem Solving Support providing the student with intelligent help on each step of problem solving - from giving a hint to executing the next step for the student.

In[43], the author mentions four techniques for adaption such as:

Adaptive Interaction occurring at the systems interface level. It provides to facilitate the users interaction with the system, without modifying the learning content itself. Examples may include the employment of alternative graphical or color schemes, font sizes, etc.

Adaptive Course Delivery optimizing the fit between course contents and user characteristics and requirements.

Content Discovery and Assembly the application of adaptive techniques in the discovery and assembly of learning material, content from potentially distributed sources / repositories

Adaptive Collaboration Support capturing adaptive support in learning processes that involve communication between multiple persons (and, therefore, social interaction), and, potentially, collaboration towards common objectives.

In this thesis, an adaptive learning system was developed which supports both adaptive course delivery and adaptive navigation.

Adaptiveness is a crucial issue in todays virtual learning environments(VLE). In [1], it is argued that VLEs are best at achieving learning effectiveness when they adapt to the needs of individual learners. VLEs should be able to identify learning needs and customize solutions that foster successful learning and performance, with or without an instructor to supplement instruction [44].In [45], it is argued that the ultimate goals of online learning environments are to achieve adaptive learning and help learners to create their own knowledge. The learning environments should also be able to support learners with learning materials that they want in a just-in-time and personalized fashion. These systems are called Adaptive Computer Assisted Instructions (ACAIs) [46] or Personalized VLEs (PVLEs) [47]. The most important issue of these systems is the customization of learning environments for diverse student communities, which has been attracting more and more attention by educational professionals and researchers [48],[49].

In [50], it is stressed that a learning system is considered adaptive if it is capable of:

- monitoring the activities of its users,
- interpreting these on the basis of domain-specific models
- inferring user requirements and preferences out of the interpreted activities,
- appropriately representing these in associated models, and
- acting upon the available knowledge on its users and the subject matter at hand, to dynamically facilitate the learning process.

2.4 Approaches of Adaptive Instructional Systems

In literature, there are different approaches used for constructing an adaptive instructional system. [37] proposes five models for creating an adaptive instruction.

1. The instructional objective and activity to achieve the objectives are fixed in this model. If the learner demonstrates the appropriate initial state of competence, then s/he can participate in the instructional activity. Otherwise, the learner is designated as a poor learner and is dropped out. If s/he does not demonstrate the achievement of the objective after the activity, then the learner is allowed to repeat the same activity or dropped out.

2. This model provides opportunity to develop appropriate initial competence for students who do not have it.

3. This model deals with different types of learning. In this model, alternative activities are available, and learners are assessed whether they have the appropriate initial competence for achieving the objective through one of the alternatives. However, there are no remedial activities for the development of the appropriate initial competence.

4. This model provides remedial activities to develop the initial competence.

5. This model allows learners to achieve different types of instructional objectives or different levels of the same objective depending on their individual need or ability. The learner is considered to be successful, if any of the alternative instructional objectives are achieved.

In[51], the author provides a well-structured review of the approaches used in adaptive instructional systems such as:

1. Macro-adaptive instructional systems

2. Aptitute-treatment interaction models

3. Micro-adaptive instructional systems

2.4.1 Macro-Adaptive Instructional Systems

Macro-adaptive instructional systems adapts instruction on macro level by allowing different alternatives in selecting only a few main components of instruction such as instructional goals, depth of curriculum content, delivery systems etc. For example, a typical pattern of teaching in a macro-adaptive instructional system includes:

- explaining or presenting specific information

- asking questions to monitor student learning

- providing appropriate feedback for the student's responses

There are several macro-adaptive instructional systems in literature.

Keller Plan It was developed at Columbia University in 1963. In this systems, the instruction is personalized for each student and has four unique features: requiring mastery of each unit before moving to the next unit; allowing self-learning pace; using textbooks and workbooks as the primary instructional means; using student proctors for evaluating student performance and providing feedback.

Audio-Tutorial System In 1971, this system was developed at Purdue University by applying audiovisual meadia, particularly audiotape. This is a tutorial-like instruction using audiotapes with other materials such as texts, slides, models etc.

PLAN A Program for Learning in Accordance with Needs(PLAN) was developed in 1967. It allows learners to select different instructional objectives and learning materials. For the selected instructional unit, the students needed to study a specific instructional unit and demonstrate a mastery before advancing to the next unit for other objectives.

Mastery Learning System This system was developed by Bloom at University of Chicago and it was a popular approach to individualized instruction. According to this system, every student achieves the given instructional objectives if sufficient time and materials are provided. This system uses both formative and summative evaluation to determine the the student's needs for more time to learn the given unit and to determine the mastery level.

IGE The Individually Guided Education(IGE)is more comprehensive macro-adaptive instructional systems. It was developed in 1965 at the University of Wisconsin. In this system, at fist the instructional objectives are determined for each student on his or her academic ability profile,which includes diagnostic assessments in reading and mathematics, previous achievements, and other aptitute and motivation data. After that, the teacher determines necessary guidance for each student, and selects alternative instructional materials. Implementation and maintenance was found to be limited.

IPI The Individually Prescribed Instruction Sytem was developed by the Learning, Research and Development Center at the University of Pittsburg in 1964. According to this system, the student is assigned to an instructional unit within a course regarding the student's performance on a placement test given before instruction.

CMI Systems The Computer-Managed Instructional (CMI) systems are able to diagnose student learning needs and prescribe instructional activities appropriate for the needs. In[51] , an example of integration of this approach and a learning management system was given. To apply CMI approach, the learning management system gives a test on different levels of instruction such as an instructional module, lesson, course, and curriculum. The learning management system evaluates each student's performance on the test and provide specific instructional prescriptions.

2.4.2 Aptitude-Treatment Interaction Models

In this approach, the specific instructional procedures and strategies are adapted regarding the specific student characteristics. It is also called aptitude-treatment interactions (ATI). Cronbach and Snow[51] defined aptitude as any individual characteristic that increases or impairs the student's probability of success in a given treatment, and defined treatments as variations in the style of instruction. There are several studies on the relationships between different aptitude variables and learning. According to ATI research findings, a few representative aptitude variables are figured out. These are explained as follows:

Intellectual Ability Snow in [51] suggested that some intellectual abilities such as verbal ability, deductive and logical reasoning, spatial relations etc. have interaction effects with instructional support. For example, more structured and less complex instruction may be more beneficial for a student with low intellectual ability, while less structured and more complex instruction,e.g discovery method, may be better for students with high intellectual ability.

Cognitive Styles Cognitive styles are characteristic modes of perceiving, remembering, thinking, problem solving, and decision making. Field-dependent versus field-independent and impulsive versus reflective styles have been considered to be the most useful in adapting instruction.

Learning Styles Learning styles are explained in section2.2 in detail. Each of learning style provides some practical implications for designing adaptive instruction. However, there is not yet sufficient empirical evidence to support the value of learning styles, and no reliable methods for assessing the different learning styles developed.

Prior Knowledge The value of prior knowledge in predicting the student's achievement and needs of instructional supports has been demonstrated in many studies. According to research findings, the higher the level of prior achievement is, the less the instructional support required to accomplish the given task is [51].

Anxiety Many studies showed that students with high test anxiety performed poorly on tests in comparison to students with low anxiety.

Achievement Motivation Although the importance of motivation with cognitive process is known, there is little research evidence available for understanding the interactions between affective and cognitive variables, particularly individual differences in the interactions.

Self-Efficacy Self-efficacy is the learner's evaluation of his or her own ability to perform a given task. Self-efficacy has an impact on people's intellectual and social behaviors, including academic achievement[52].

2.4.3 Micro-Adaptive Instructional Systems

This approach is aimed to diagnose learner's specific learning needs during instruction and to provide instructional prescriptions for the needs. It is designed to guide the learner's ongoing learning process throughout the instruction, and therefore the diagnosis and prescription are often continuously performed from the analysis of the student's performance on the task. This ongoing diagnosis of the learner's learning process is the point where this approached differs from the other two approaches. One-on-one tutoring is a typical example of micro-adaptive instruction. In tutoring, the tutor selects the most appropriate information to teach based on his or her judgement of the student's learning ability, including prior knowledge, intellectual ability and motivation. Then, the tutor continuously monitors and diagnoses the student's learning process and determines the next instructional actions[51].

In[51], it is explained that micro-adaptive instructional systems have been developed through a series of different attempts beginning with programmed instruction to the recent application of artificial intelligence (AI) methodology for the development of intelligent tutoring systems(ITS).

Programmed instruction is a technique for presenting a subject matter to a student who can work through it at his own learning speed. It consists of statements and tests, which direct the student to new statements depending on his pattern of errors. This technique provides some important implications for the development of more sophisticated instructional strategies made possible by the advance in computer technology. Skinner[51] is considered as the pioneer of programmed instruction. Skinner designed a teaching machine to arrange contingencies of reinforcement in school learning. The instructional program format used in the teaching machine had the following characteristics:It was made up of small, relatively easy-to-learn steps; the student has an active role in the instructional process; and positive reinforcement was given immediately following each correct response[51].

The computer technology made possible to design and develop micro-adaptive instructional models, because the micro-adaptive model is more sophisticated and difficult to manage. The micro-adaptive instruction uses the temporal nature of learner abilities and characteristics as a major source of diagnostics information. So, it has a dynamic nature. It includes more variables than the other approaches, mostly represented by quantitative measures.

There are several studies on micro-adaptive models. These are mathematical model, trajectory model, Bayesian probability model, structural and algorithmic approach. The explanations of each model is given the following section.

Mathematical Model Mathematical learning theory is an attempt to describe and explain behavior in quantitative terms. Atkinson[51] discusses the problem of optimizing instruction. The main principles of the mathematical model are as follows:

1. It is possible to develop an optimal instructional strategy for a given individual provided that a detailed model of the learning process is available

2. Optimal learning performance can be achieved by giving each individual sufficient time to learn

Atkinson outlined four possible strategies:

1. maximize the mean performance of the whole class

2. minimize the variance in performance for the whole class

3. maximize the number of students who score at grade level

4. maximize the mean performance for each individual

There are some criticisms of the mathematical adaptive instructional models. One is that the learning process in the mathematical model is oversimplified. The other is that there is a substantial amount of student and content data accumulated in order to make estimations for instructional diagnosis and prescription[51].

Trajectory Model: Multiple Regression Analysis Approach In this model, numerous variables are included with the use of a multiple regression technique to understand what may be more powerful and precise predictive base than is obtained by considering a particular variable alone. Hansen et.al used this model to develop and micro-adaptive model. The procedures that the author applies were as follows[51]:

- Learning and test materials were prepared. The data was collected from

  - two measures of personality measures (locus of control and trait anxiety),
  - one measure of general aptitude related to the math and verbal
  - one measure of general aptitude of a subject familiarity(pretest).

  After completing the pretest, the subject was given the programmed manual and task instructions, then the student work on the manuals and tool the posttest. The measures of the four entry variables and the posttest score provides the prediction on the formulation of adaptive grouping.

- By using the cluster analysis technique, students with similar characteristics are clustered in one of a small number of mutually exclusive groups.

- The new students receiving the adaptive treatments were classified into one of the groups by discriminant analysis. This is a method used to seek the linear combination of variables that will maximize the difference between the groups relative to the difference within the groups.

- Multiple regression analysis was used to derive differential predictions about the number of instructional items to assign to the student.

- The initial prescriptions derived from the group characteristics were redefined during instruction on the basis of the student's performance on the immediately preceding rule posttest.

There are some limitations of this model. It does not seem to be a very useful adaptive instructional strategy. It is limited to the adaption of instructional amount. Also, unless the number of students to be taught is large, this approach cannot be effective since the establishment of the predictive database in advance requires a considerable number of students, and this strategy cannot be applied to those students making up the initial predictive database[51].

Bayesian Probability Model Two-step approach is used in the Bayesian probability model to adapt instruction to individual students. In the first step, the initial assignment of the instructional treatment is made on the basis of pre-instructional measures. Then, in the second step, the treatment prescription is continuously adjusted according to student on-task performance data.

The Bayes' theorem of conditional probability predicts the probability of mastery of the new learning task from student pre-instructional characteristics and then continuously update the probability according to the on-task performance data(Rothen and Tennyson, 1978; Tennyson and Chrstensen, 1988 cited in [51]).

The computer technology provides a powerful means to use all of these adaptive instructional approaches in online learning settings. This study is an attempt to build a computer system which provides micro-level adaptive instruction to be applied in learning management systems.

2.5 Review of Adaptive Learning Systems

In literature, there are many adaptive learning systems supporting different adaptive techniques, having implemented with different technologies and having different methodologies. These systems includes iWeaver[8], INSPIRE[9], Arthur[11], CS388[12], AEC-ES[13], InterBook[3], ACE[5], ELM-ART II[6],
ILESA[7], etc. Each of these adaptive systems is explained separately by focusing on the description, the adaptation techniques supported, the technologies used, and the pedagogical methodologies applied.

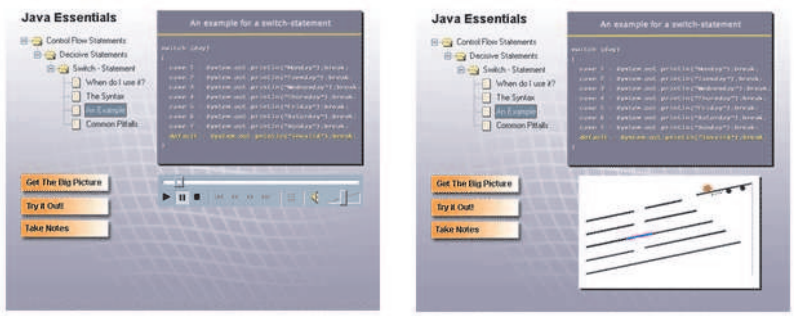
2.5.1 iWeaver

Description iWeaver is an interactive web-based adaptive learning environment. The best matching combination of media experiences and learning tools are calculated and recommended to the learner (Figure2.4).

Adaptation Techniques It supports only adaptive presentation, navigation, link hiding and link ordering.

Technology JSP, JavaBeans, Tomcat, MySQL

Methodology iWeaver uses the Dunn and Dunn[29] learning style model while modeling the learner. The learner completes a questionnaire and initialize his/her profile. When the learner enters the learning environment and is provided with different contents, two out of the total number of four available contents each for visual text, visual pictures, tactile kinaesthetic and auditory learners. These choices are tailored towards the perceptual domain of the Dunn and Dunn learning styles model. According to the learner responses and interactions with the environment, the learner profile is updated on the fields including navigational choices, usage pattern of learning tools, etc. To adapt, the learner model is refined and compared with the content representation model.



(a) b)

Figure 2.4: iWeaver Interfaces: a) the auditory learner interface b)the visual learner interface[8]

2.5.2 INSPIRE

Description INSPIRE is an adaptive intelligent system for personalized instruction in a remote environment. It adapts the lessons according to the learner's knowledge level, learning style and follows his/her progress.

Adaptation Techniques INSPIRE involves curriculum sequencing, adaptive navigation, adaptive presentation techniques.

Technology ASP, SQL Server, ActiveX Data Objects(ADO), IIS Web Server

Methodology It is based on Honey and Mumfor's[15] learning style model. The learner model involves general information,learning style and knowledge. The knowledge of the learner on the lesson is categorized as *Insufficient*, *Rather Sufficient*, *Almost Sufficient*, Sufficient. According to the knowledge level, the learner's access to appropriate content which are categorized as *Remember*, *Use*, *Find*. For example, if the knowledge level of the learner has been evaluated as *Insufficient* on a number of outcome concepts, then, s/he has to study the educational material of the Remember level on these outcome concepts and their entire prerequisite ones.

2.5.3 Arthur

Description Arthur is a web-based instruction system that provides adaptive instruction. It makes use of several different styles of instruction from different instructors and provides them to each learner.

Adaptation Techniques It supports adaptive presentation.

Technology Java Applet, Java Sockets, Intelligent FAQ, Knowledge Base, SQL Database

Methodology Student learning style is detected and tuned by means of case-based reasoning techniques. The instructors from the same discipline work together and construct the concept map of the course. When the learner enters Arthur, the first concept of one of the course modules from the instruction pool is delivered by chance. When the learner takes a short evaluation quiz, the concept becomes terminated. The student must pass each section with a score of eight percent or better in order to continue within the current course module. When a learner passes a concept, then, Arthur assumes that the instruction style used in that section matches the learner's learning style. When the learner fails in the questions of the quiz, the system uses this information while classifying the future learners. The system summary of the tool is summarized in Figure 2.5.

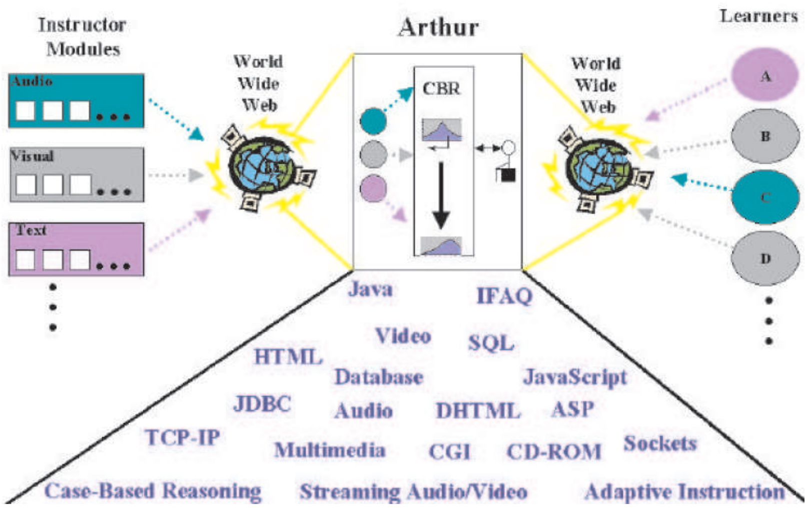


Figure 2.5: System Summary of Arthur[11]

2.5.4 CS383

Description CS383 is a computer systems course. It provides a vast array of media elements to support the transfer of information. It provides: 143 audio files, 63 graphic files, 57 digital movies, instructor slideshows for every lesson, lesson objectives, notetaking guides, a student legacy system with over 471 student papers and slideshows from previous semesters, and 300 pages of course hypertext with 178 cross references, 678 terms with pop-up definitions, and 600 terms that students can search.

Adaptation Techniques It provides adaptive presentation.

Technology CGI

Methodology It is based on the Felder-Silverman[14] learning style model.In the study, the learning style dimensions and the adaptive strategies are figured out for some of learner actions. For example, for the learner's view lesson objectives action, it is explained that:

- Lesson objectives particularly address global, verbal, and intuitive learners.
- Global learners appreciate the lesson overview before the details are presented. Students can also scan the objectives of different lessons or the entire course and determine the "big picture."
- Lesson objectives also address verbal learners. Verbal learners prefer words, either oral or written, as their preferred method of learning.
- Finally, intuitive learners respond to concepts and the lesson objectives detail the main concepts of the course.

The learner profile is initialized through a survey. The appropriate content is determined by using the learner profiles. Certain media are inherently appropriate to different learning styles. For example, slide-shows, graphics, and digital movies clearly appeal to visual-learners while the course hypertext with its text-based, hierarchical, presentation of material appeals to verbal, sequential learners. In the study, each course tool was rated on a scale from 0 to 100 to determine the amount of support for each learning style. This rating was combined with the student profile to produce a unique ranking of each media type from the perspective of the student's unique profile. This ranking will differ from course to course depending on the course and media content. Different courses, media, and instructors will result in different tool ratings[12].

2.5.5 AEC-ES

Description AEC-ES is an adaptive educational system based on cognitive styles. According to[13], the cognitive style is the form of cognitive activity such as thinking, perceiving and remembering, learning style, on the other hand, is broader concept including cognitive along with affective and physiological styles.

Adaptation Techniques AEC-ES system involves adaptive content delivery, adaptive presentation and adaptive navigation support.

Technology ASP, DHTML and JavaScript

Methodology AEC-ES functions on top of the most well known cognitive styles, field dependence/independence (FD/FI). The field-dependent learners are supported with navigational support tools such as concept map, graphical path indicator and advanced organizer. The field-independent learners are given more control by providing a menu from which they can proceed with the course in any order. In AEC-ES, the learner was modeled using the following items[13]:

- Personal Profile
  - username
  - password
- Cognitive Profile
  - Cognitive Style
  - Program Control
  - Learner Control
  - Advance Organizer
  - Post Organizer
  - Graphics Path Indicator

- Knowledge Profile
  - Concept 1
    - * Unknown
    - * Known
    - * Learned
    - * Well-Learned
  - Concept 2
  - ...

2.5.6 InterBook

Description InterBook is a tool for delivering adaptive textbooks on the World Wide Web. Interbook allows the creation of adaptive electronic textbooks based on hierarchically structured MS-Word files. Courses compiled with Interbook provide individual guidance to students by annotating the navigational structure of the hypertext due to the users learning progress, by generating individually learning paths and by personalized embedding of exercises (Figure2.6).

Adaptation Techniques It supports adaptive annotation and adaptive presenation technology.

Technology LISP, Common Lisp Hypermedia Server CL-HTTP, HTML and RTF.

Methodology The InterBook approach uses two kinds of knowledge: knowledge about the domain being taught (represented in the form of a domain model) and knowledge about the students (represented in the form of individual student models). Interbook uses domain concepts which are elementary pieces of knowledge for the given domain.

The documents in Interbook are units from indexed electronic textbooks. The knowledge about the learner involves the grades for each concept. The grade might be *Beginner-Knowledge*, *Intermediate-Knowledge*, *Expert-Knowledge*, and *No-Knowledge*.

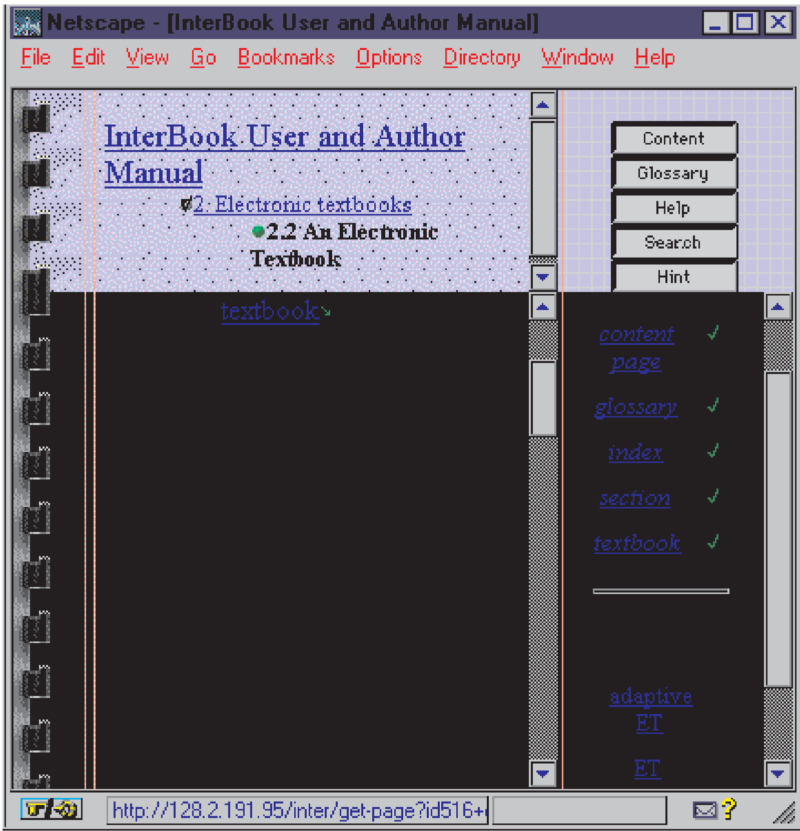


Figure 2.6: InterBook[3]

2.5.7 ACE

Description ACE(Adaptive Courseware Environment) It is a WWW-based tutoring framework which combines methods of knowledge representation, instructional planning, and adaptive media generation to deliver individualized courseware via the WWW.

Adaptation Techniques It supports curriculum sequencing, adaptive presentation and adaptive navigation.

Technology Java, HTML

Methodology ACE is based on a domain model of the subject matter, a pedagogical model on how to teach a curriculum, and learner modeling on different levels, e.g., preferences, interests, and knowledge. The domain the model is built on a conceptual network of learning units. Learning units can either be sections or concepts. The pedagogical model represents the instructors knowledge of how to teach units. It consists of both teaching strategies and diagnostic knowledge. There are certain rules implemented in ACE to select appropriate teaching strategies based on learner characteristics such as if a learner is not familiar with prerequisites of the current unit, then it gives warning, introduction and hyperlink to prerequisite testing as a teaching strategy. The diagnostic knowledge is obtained from several types of questionnaires and user dialog. The learner model consists of the learner settings, the knowledge model, and the interest model.

The learner model consists of three main parts:

- The learner settings stores information about the learner preferences for language, media, interface settings, personal annotations, and current courseware booked by the student.
- The knowledge model consists of the units a learner worked on. Learned units have confidence values depending on the experiences of a learner with a unit. They defined three confidence values such as tested confidence when the leaner takes a test, requested confidence when learner request information of any type, and inferred confidence when the system infer some knowledge through learner actions.
- The interest model contains the interest clusters a learner is interested in and dynamically builds hypotheses about the learn-

ers interests.

A sample screeon shot of ACE is given in Figure 2.7.

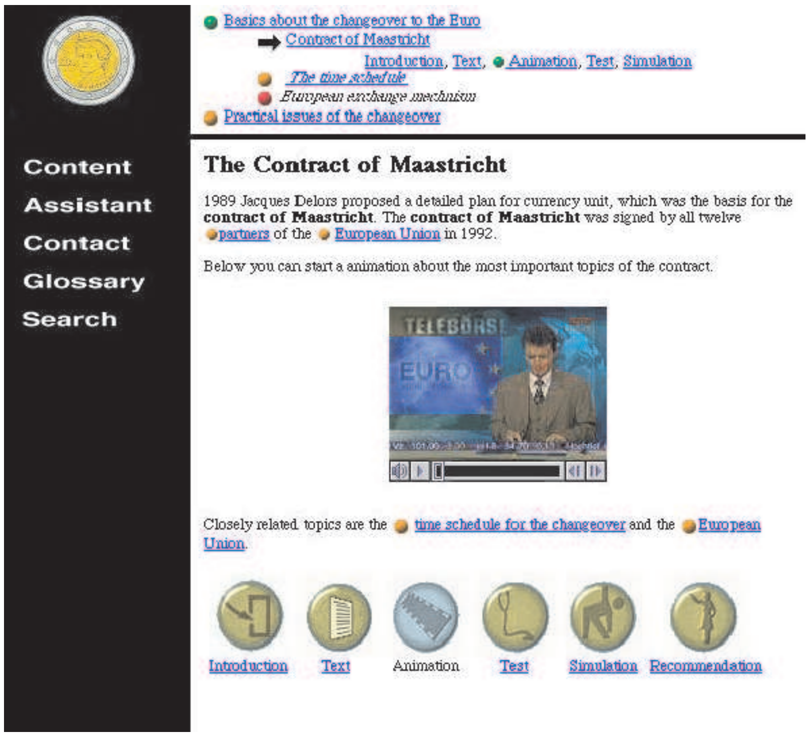


Figure 2.7: A sample screen from presentation in ACE[5]

2.5.8 ELM-ART II

Description ELM-ART (The Episodic Learning Model: The Adaptive Remote Tutor) II, an intelligent interactive educational system to support learning programming in LISP. It provides all learning material online in the form of an adaptive interactive textbook. It makes use of ELM-PE, a knowledge-based programming environment for learning LISP (Figure2.8).

Adaptation Techniques It supports adaptive navigation, curriculum sequencing, intelligent solution analysis, and problem solving support

Technology Common LISP Hypermedia Server CL-HTTP, LISP, CGI, HTML

Methodology ELM-ART II represents knowledge about units to be learned with the electronic textbook in terms of a conceptual network[42]. Units are organized hierarchically into lessons, sections, subsections, and terminal pages.

It has episodic learner modeling (ELM). It means that it stores knowledge about the learner in terms of a collection of episodes. In the sense of case-based learning, such episodes can be viewed as cases[42]. To construct the model, the code produced by a learner is analyzed in terms of the domain knowledge on the one hand and a task description on the other hand. This cognitive diagnosis yield to a derivation tree of concepts and rules the learner might have used to solve the problem. These concepts and rules are instantiations of units from the knowledge base. The episodic learner model is made up of these instantiations.

The content that best fits the current learning situation is chosen on the basis of the individual episodic learner model. Episodic learner modeling is well suited for diagnosing complete and incomplete solutions to problems and giving individualized help.

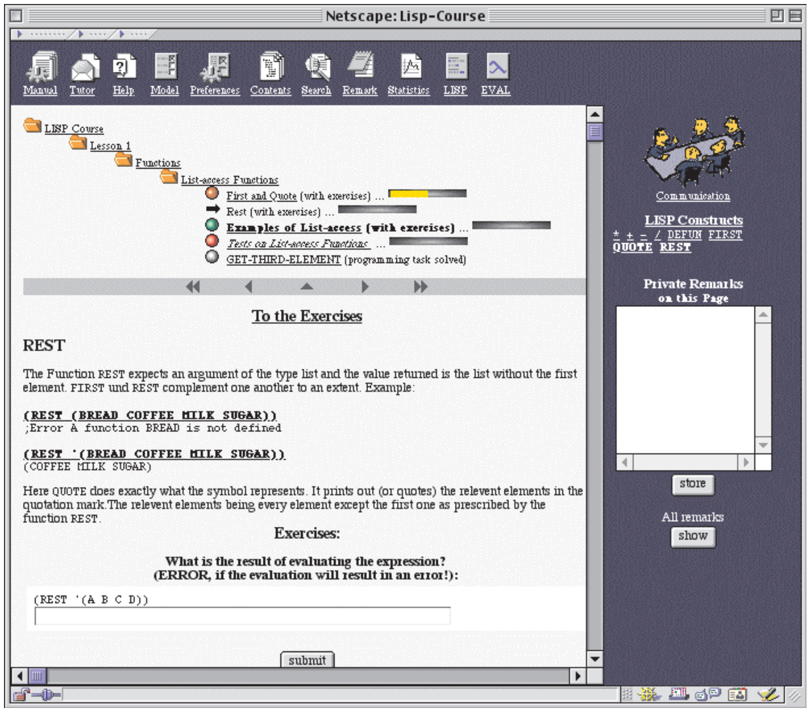


Figure 2.8: Example of a textbook page with an exercise using ELM-ART[6]

2.5.9 ILESA

Description ILESA is a web-based intelligent learning environment developed to help teachers in the task of teaching the Simplex Algorithm. The Simplex Algorithm was developed by Dantzig in 1940 and is an iterative procedure to solve linear programming problems, that consists in finding the optimum of a linear function subject to a number of linear constraints. The interaction with ILESA will start when the teacher considers that the student has the necessary theoretical background to begin to solve problems. At first, ILESA will propose easy problems, which will increase in difficulty as the student exhibits more competence in solving them, with the goal of being capable of solving any linear programming problem. It is also able to detect mistakes, inform the student and offer help to fix them[7].

Adaptation Techniques adaptive presentation and problem solving support

Technology Java (version 1.1.4), HTML

Methodology From a pedagogic point of view, the system can be classified in the category of coached problem solving systems. The systems has the following modules:

The Expertise Module The domain of this system is an iterative algorithm, so the expert module is a linear programming problems solver.

Student Diagnosis Module In the study, they defined 22 skills for solving Simplex Algorithm. The student model consists of an array of numbers between 0 and 5 (each number is associated to one of the these basic skills and represents the confidence in the student to have developed that skill) that is updated according to the student's answers, and a number that indicates which type of problem must be posed to the student.

Problem Generator This module is used to generate an unlimited number of problems for the student, providing always the adequate type and level of problem.

Instructional Module This module manages the learning strategy to control the pedagogic functioning of the system. For example, if the student has to select the entering vector from a choice of three vectors, ILESA will ask him to explain his selection, to discourage students from guessing.

The Student Interface This module allows the student to interact with the system.

CHAPTER 3

Agent Technology

Agents are being used in an increasingly wide variety of applications, ranging from comparatively small systems such as email filters, to large, open, complex, mission critical systems such as air traffic control[53]. Agent technology is a new paradigm for developing software applications in education domains.

This chapter introduces some basic concepts such as intelligent agents, agent types, and agent application domains. A review of the pedagogical agents in education application domain are given at the end of the chapter. The aim of this chapter is to help the reader to understand why agent technology is seen as a fundamentally important new tool for building adaptive learning environments.

3.1 Intelligent Agents

Nwan and Ndumu [54] defines agent as a component of software and/or hardware that is capable of acting exactingly in order to accomplish tasks on behalf of its user.

Ferber [55] proposes the following definition that Something can be called an agent if it is a physical or virtual entity that:

- is able to act in an environment,
- can communicate directly with other agents,
- is driven by a group of tendencies [...],
- has its own resources,
- is able to perceive its environment (but in a limited way),
- has only a partial representation of this environment (and possibly none),
- has skills and offers services,
- may be able to reproduce,
- whose behavior aims to satisfy its objectives, by taking account of the resources and skills that it possesses, and according to its perception, of its representations and the communications that it receives.

According to [56], agents are described as programs, used extensively on the Web, that perform tasks such as retrieving and delivering information and automating repetitive tasks.

According to Croft[57], agent is one that is authorized to act for another. Agents possess the characteristics of delegacy, competency, and amenability. Delegacy means discretionary authority to autonomously act on behalf of the client. Actions include making decisions, committing resources, and performing tasks. Competency is the capability to effectively manipulate the problem domain environment to accomplish the prerequisite tasks. Competency includes specialized communication proficiency.

Amenability is the ability to adapt behavior to optimize performance in an often non-stationary environment in responsive pursuit of the goals of the client. Amenability may be combined with accountability.

Software agent is an artificial agent which operates in a software environment [57]. Intelligent Software Agent (ISA) is a software agent that uses Artificial Intelligence (AI) in the pursuit of the goals of its clients [57].

Artificial Intelligence is the imitation of human intelligence by mechanical means. Clients, then, can reduce human workload by delegating to ISAs tasks that normally would require human-like intelligence.

Many researchers that formerly referred to their work as AI are now actively engaged in *agent technology.* Thus the word *agent* by itself generally connotes ISAs in the terms of the present-day research community.

Delegacy for ISAs is far more absolute. ISAs have the capability to generate and implement novel rules of behavior which human beings may never have the opportunity or desire to review. As ISAs can engage in extensive logical planning and inferencing, the relationship of trust between the client and the agent is or must be far greater, especially when the consumption of client resources is committed for reasons unexplained or multiple complex operations are actuated before human observers can react.

Competency as practiced by ISAs adds higher order functionality to the mix of capabilities. In addition to communicating with their environment to collect data and actuate changes, ISAs can often analyze the information to find non-obvious or hidden patterns, extracting knowledge from raw data. Environmental modes of interaction are richer, incorporating the media of humans such as natural language text, speech, and vision.

Amenability in ISAs can include self-monitoring of achievement toward client goals combined with continuous, online learning to improve performance. Adaptive mechanisms in ISAs mean that they are far less brittle to changes in environment and may actually improve. In addition, client responsiveness may go so far as to infer what a client wants when the client himself does not know or cannot adequately express the desired goals in definitive terms.

According to [58], on the Internet, an intelligent agent (or simply an agent) is a program that gathers information or performs some other service without your immediate presence and on some regular schedule. Typically, an agent program, using parameters you have provided, searches all or some part of the Internet, gathers information you're interested in, and presents it to you on a daily or other periodic basis. An agent is sometimes called a bot (short for robot).

Other agents have been developed that personalize information on a Web site based on registration information and usage analysis. Other types of agents include specific site watchers that tell you when the site has been updated or look for other events and analyst agents that not only gather but organize and interpret information for you.

3.2 Agent Types

Agent is an umbrella term that covers a range of other more specific agent types. In [54], the authors mention six type of agents. These agent types and their brief descriptions are as follows:

3.2.1 Collaborative agents

Collaborative agents interact with each other to share information. They emphasize autonomy and cooperate with other agents in order to perform certain tasks for their owners in open and time-constrained multi-agent environments[54]. They negotiate in order to reach an agreement. Collaborative agents help to solve the large scale problems which are too large for a centralized single agent. They are also seemed to be a solution to inherently distributed problems. An examples for collaborative agents includes ADEPT[59] used in business process re-engineering[54].

3.2.2 Interface agents

Interface agent is the personal assistant collaborating with the user. Collaborative agents collaborates with other agents, but interface agents collaborate with the user. Interface agents make less work for the end user and application developer and it can adapt, over time, to its user's preferences and habits[54]. In literature, there are several interface agents use such as Calendar Agent[60] as assisting its user in scheduling meetings and learning the preferences and commitments of its users, and Letizia[61], a keyword and heuristics-based search agent assisting in web browsing.

3.2.3 Mobile agents

Mobile agents are software agents that are capable of roaming wide area networks, moving to the foreign hosts, performing tasks on there and returning home having performed the responsibilities set[54]. Mobile agent frameworks are currently rare, however, due to the high level of trust required to accept a foreign agent onto one's data server. Sony's Magic Link PDA is an example mobile agent product. It assists in managing the user's email, fax, phone, and pager as well as linking the user to TeleScript enable messaging and communication services. TeleScript is an interpreted object-oriented and remote programming language[54].

3.2.4 Information/Internet agents

Information agents are responsible for managing, manipulating, or collating information from many distributed sources. After WWW, it becomes very difficult to manage information. These agents meet the needs for information management issues[54]. Information agents can also be a mobile agent when necessary. An example for Information agents is Jasper agent[62]. It is used to store, retrieve, summarize ad inform other agents of information useful to them found on WWW.

3.2.5 Reactive Agents

Reactive agents are also known as autonomous agents. In [54], these agents are described in a way that these agents do not posses internal, symbolic models of their environments; instead they respond in a stimulus-response manner to the present state of the environment in which they are embedded. The motivation behind the reactive agents it that they would be more robust and fault-tolerant than other agent-based systems.

3.2.6 Hybrid Agents

Hybrid agent is a single agent which combines two or more different agent's philosophies to gain the maximum benefit.

3.3 Agent Application Domains

For any new technology to be considered as useful in the computer marketplace, the agents must offer one of two things:

1. the ability to solve problems that have been beyond the scope of automation, either because no existing technology could be used or because it was considered too expensive

2. the ability to solve problems that can already be solved in a significantly better (cheaper, more natural, easier, more efficient, or faster) way.

There are several application domains that agents are applied to solve problems. These are[54]:

- Industrial Applications
- Commercial Applications
- Medical Applications
- Entertainment

In industrial applications, agents are mostly used in process control, manufacturing, and air-traffic control. In commercial application, the information management, electronic commerce, and business process management are the areas in which agents are applied. Kashbah[63] is an example for an agent working in the area of e-commerce. ADEPT[59] is an example agent used in this application domain. Medical informatics is a major growth area in computer science. Agents are used in the areas of health care and patient monitoring. The other field for using agents is entertainment. The agents have an obvious role in computer games, interactive theater, and related virtual reality applications: such systems tend to be full of semi-autonomous animated characters, which can naturally be implemented as agents [54].

3.4 Agents in Education, Pedagogical Agents

Another application domain in which agents can solve many problems is the field of distance education. There are several difficulties for the instructors to provide an effective teaching in online learning settings such as:

- changing teaching styles to that of a mentor, tutor, and facilitator when necessary,
- meeting the needs of distance students without face-to-face contact,

- changing the course content to accommodate diverse student needs and expectations.
- giving immediate feedback,
- assessing student performance using different assessment strategies.

It is aimed to use agents to solve these problems and help both online instructors and learner to overcome the problems of distance education.

3.4.1 Definition of Pedagogical Agents

The agents in education are defined as pedagogical agents. Pedagogical agents are autonomous agents that support human learning, by interacting with learning in the context of interactive learning environments [64].

Pedagogical agents are autonomous agents that support human learning by interacting with students in the context of interactive learning environments. They extend and improve upon previous work on intelligent tutoring systems in a number of ways. They adapt their behavior to the dynamic state of the learning environment, taking advantage of learning opportunities as they arise. They can support collaborative learning as well as individualized learning, because multiple students and agents can interact in a shared environment[64].

Jafari[65] conceptualizes three types of pedagogical agents to assist teachers and students and to expand the capabilities of CMS into an intelligent teaching and learning environment. These intelligent agents and their descriptions are as follows:

- **Digital Teaching Agent** is a personal agent that can be configured by its owner, the human instructor. The responsibilities of this agents are given in the following:

- know if and when students worked on assignments, for how long, or what types of collaboration they used,
- dynamically aware of student participation in a course,
- assist a discouraged student before s/he drops out,
- assist course instructor with course operation and maintenance.

• **Digital Tutor** assist students with specific learning needs.

- act as a smart search engine, finding specific resources to solve learning needs,
- depending on the level of its sophistification, it could "learn" and become more expert and useful as it provides more assistance to a student and receives more feedback,
- knows students strengths and weaknesses on a learning objective,
- act as a communication agent, dynamically show the list of online students within a course.

• **Digital Secretary** assists students and instructor in various logistical and administrative assistant needs.

- One task might be "out-of-office" e-mail notification,
- Digital secretary offer more intelligent and sophisticated services than the out-of-service agent, for example: send a different auto-response e-mail to only those students taking a specific undergraduate course or those in the course that meets in the evening.

3.4.2 Review of Pedagogical Agents

There are several pedagogical agents developed in literature. This section provides a review of these agents.

3.4.2.1 PPP Persona

Description It is developed by Andr, Rist and Mller at the German Research Centre for Artificial Intelligence (DFKI). PPP Persona is an animated pedagogical agent for interactive WWW presentations. It can be used for showing, explaining, and verbally commenting textual and graphical output on a window-based interface. The persona appears in many forms. Currently there are two cartoon figures and three 3D models. The persona guides the learner through Web-based material using presentation acts (e.g. pointing) to draw attention to elements of the Web pages, and provide commentary via synthesized speech. The PPP system generates multimedia presentation plans for the persona to deliver (Figure3.1).

Adaptation Techniques Adaptive presentation

Technology Java, HTML

Methodology PPP persona executes this plan adaptively, modifying it in real-time based on user actions such as repositioning the agent on the screen or asking questions. It follows the client/server paradigm, i.e, some client applications can send requests for executing presentation tasks to the server. However, to achieve a lively and appealing behavior of the animated agent, the server autonomously performs some actions, eg. to span pauses or to react immediately to user interactions. While the user views the presentation, the agent can comment on particular parts and highlight them through pointing gestures. The repertoire of the personas presentation includes gestures expressing approval or disapproval, warning or recommendation, etc.

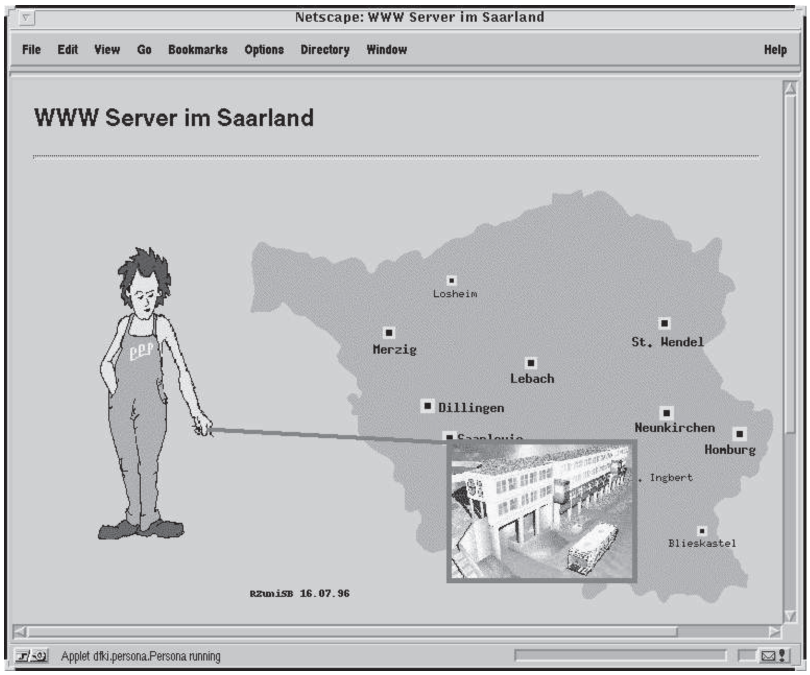


Figure 3.1: PPP Persona[17]

3.4.2.2 The SQL Tutor

Description SQL-Tutor is an ITS for assisting students in learning the database query language, SQL. The system is aimed at upper-level undergraduates. It is developed as a guided discovery learning environment that provides facilities to verify students solutions and assist them in solving problems, if required (Figure3.2).

Adaptation Techniques Problem Solving Support

Technology Common Lisp (CL) HTTP server, HTML

Methodology SQL-Tutor consists of a user interface, pedagogical module and a student modeller. The pedagogical module (PM) selects problems for the student and generates appropriate feedback. When a solution to a problem is submitted, PM sends it to the student modeller. The student modeller checks whether the solution is correct or incorrect and updates the student model. SQL-Tutor is based on a constraint-based modelling(CBM) that focuses on student errors. The domain knowledge in CBM is represented as constraints and is used to identify errors. The assumption of CBM is that the diagnostic information is not in the sequence of student actions, but is in the final state. The level of feedback determines the amount of information provided to the student. The system provides six levels of feedback: positive/negative, error flag, hint, partial solution, all errors and complete solution.

Figure 3.2: User Interface of SQL-Tutor (Web version)[66]

3.4.2.3 ADELE (Agent for Distance Learning Environments)

Description ADELE uses a client-side ITS and it includes a graphical agent. It was constructed on a web-delivered trauma care course. The students in that course had to learn a correct procedural sequence for assessing an injured patient. The agent notifies the current status of the patient by playing sample audio tracks such as breath sounds. When students does not follow the steps in the procedure, ADELE interrupts and warn the student about the his/her incorrect action. The agent can then prompt the student to continue, using the correct procedure. When the procedure is complete, students can query ADELE for further comments. A sample interface of ADELE is given in Figure3.3.

Adaptation Techniques It supports adaptive presentation.

Technology Java Applet, HTML, Simulation authoring tools such as VIVIDS and Emulteks RAPID

Methodology It is responsible for monitoring student, recording student actions, adapting courseware presentation as needed and reporting student performance to the central server at the end of the session. ADELEs system consists of two main components: the pedagogical agent and the simulation. The pedagogical agent consists further of two sub-components, the reasoning engine and the animated persona. ADELE has been adopted for a case-based clinical diagnosis application, where this is used to highlight interesting aspects of the case, and monitor and give feedback as the student works through a case.

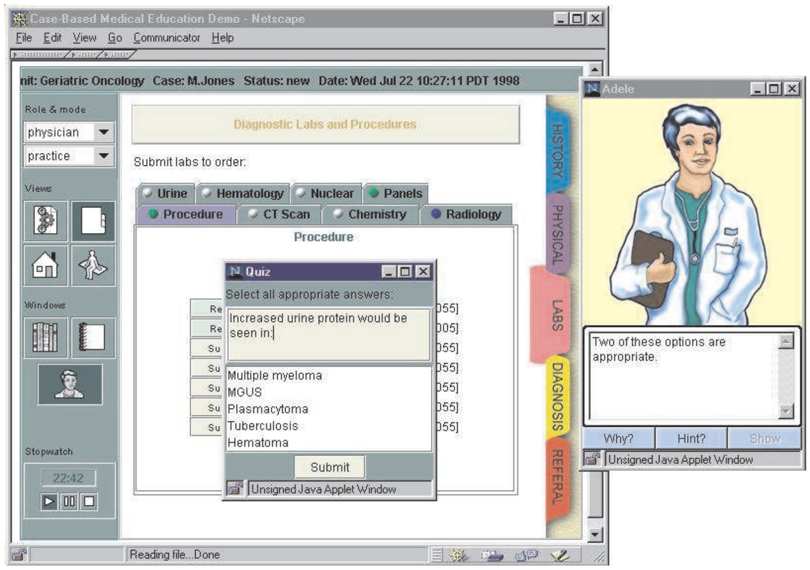


Figure 3.3: ADELE instructs a student to answer a quiz after the student elects a urine dipstick test [8]

3.4.2.4 Herman the Bug

Description The Design-A-Plant is a knowledge-based learning environment for the domain of botanical anatomy and physiology. Herman the Bug is an animated pedagogical agent living in this environment (Figure3.4).

Adaptation Techniques Adaptive Navigation

Technology Unknown

Methodology Herman observes students actions as they build plants that can thrive in a given set of environmental conditions and provides explanations and hints. In the process of explaining and hinting, Herman performs various actions, such as walking, flying, swimming, teleporting, etc.

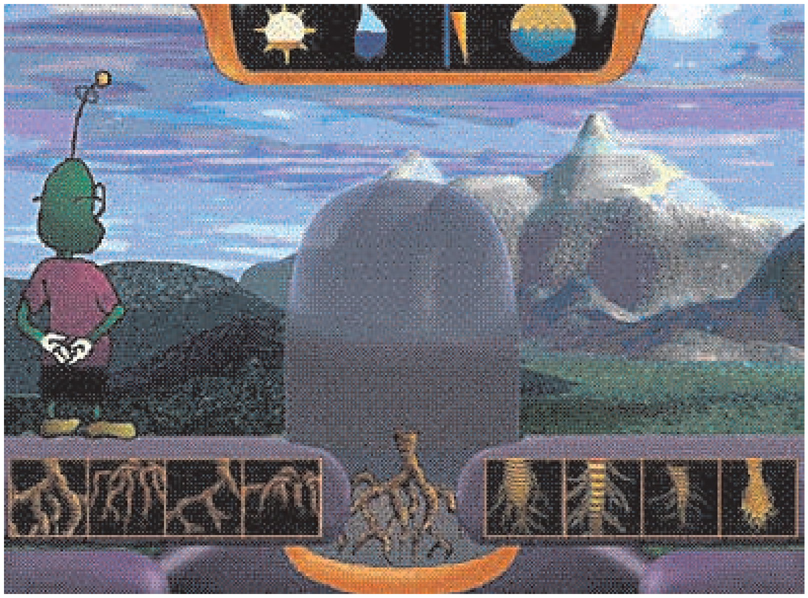


Figure 3.4: Herman the Bug[67]

3.5 Agent Development Environments

There are several java-based agent development environments such as ABLE[68], AgentBuilder[69], Aglets[70], FIPA-OS[71], JADE[19], JATLite[72], etc. In this study, JADE was chosen as the agent development environment. The brief descriptions about JADE and other agent development environments are provided below.

ABLE (Agent Building ad Learning Environment)[68] is a Java framework, component library, and productivity tool kit for building intelligent agents using machine learning and reasoning. The ABLE research project is made available by the IBM T. J. Watson Research Center. ABLE provides a set of reusable JavaBean components, called AbleBeans, along with several flexible interconnection methods for combining those components to create software agents. It provides a graphical user interface-based interactive development environment [73]. It is possible to obtain information about ABLE at www.alphaWorks.ibm.com/tect/able.

AgentBuilder[69] is an integrated software toolkit that allows software developers to quickly develop intelligent software agents and agent-based applications. It is developed by Reticular Systems Inc. There are two main products int AgentBuilder toolkit such as Agent Builder Lite and AgentBuilder PRO. The former is suitable for building single-agent standalone applications and small agencies. The latter one has all features of Lite plus an advanced suite of tools for testing and building multi-agent systems[69]. This toolkit uses a high-level, agent-oriented programming language and provides a suite of graphical programming tools for configuring agents and specifying their behaviors. AgentBuilder is intended to enable developers who have no artificial intelligent background to build intelligent applications. In addition to agent-level development and debugging tools, it provides a set of graphical project management and domain analysis tools[73]. It is possible to find additional information on AgentBuilder's Web site at www.agentbuilder.com.

Aglets[70] are Java objects that can move from one host on the Internet to another. Aglets are developed by IBM. That is, an aglet that executes on one host can suddenly halt execution, dispatch itself to a remote host, and resume execution there. When the aglet moves, it takes along its program code as well as its data. IBM turned over the source to the open source community. IBM provides API documentation and the Aglets software development kit[73]. One can refer to www.trl.ibm.co.jp/aglets.

FIPA-OS[71] is an environment developed at Nortel Networks. There is an agent communication language standard, named FIPA[74]. FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies. FIPA, the standards organization for agents and multi-agent systems was officially accepted by the IEEE as its eleventh standards committee on 8 June 2005. FIPA specifications represent a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that they can represent[73][71]. FIPA-OS is an open-agent platform that supports communication using FIPA. It provides the set of platform services that are specified in the FIPA agent standards, including an agent management system for life-cycle management, a director facilitator or yellow pages service, and an agent communication channel for FIPA-compliant messaging and interaction protocols. For more information about FIPA-OS, one can refer to http://fipa-os.sourceforge.net

JADE[19] is a FIPA-compliant java agent development environment developed at CSELT S.p.A in Torino, Italy. It is developed to create multi-agent systems applications. JADE provides a set of tools for debugging and deploying distributed agents. It provides a set of agent services including an agent-naming service, transport protocols, yellow pages service, and interaction protocols that are FIPA compliant. A GUI is provided for remote monitoring and control of agents that are running on the JADE agent platform[73][19]. Additional information can be found in http://sharon.cselt.it/projects/ jade/home.htm.

JATLite[72] is Java Agent Template Lite involving a set of lightweight Java packages being developed at Stanford University can be used to build multi-agent systems. JATLite (Java Agent Template, Lite) is a package of programs written in the Java language that allow users to quickly create new software "agents" that communicate robustly over the Internet. The most unique feature of the JATLite is the agent infrastructure packaged with it. Traditional agent systems use an Agent NameServer (ANS) for making connections between agents. With the JATLite infrastructure, all agents make a single connection to the AMR. The AMR forwards all agent messages by name to the last known IP address. Further, just like an email system, the AMR buffers all messages and saves them until the receiving agent acknowledges receipt with a delete message to the AMR[72]. For more information, one can refer to http://java.stanford.edu/java-agent/html.

CHAPTER 4

A Conceptual Framework for Adaptive Learning Systems

In this study, a conceptual framework for adaptive learning systems is given. The framework is based on the idea that the adaptiveness is the best matching between the learner profile and the course content profile. The learning styles of learners and content type of learning material are used to match the learner to the most suitable content.

The conceptual framework involves learner profile, course content profile, the matching strategies between learner and course content profiles, and the initialization and update strategies of the profiles. The framework is based on the idea that the effectiveness of the adaptiveness is highly dependent on how much we know about the learner and how much the available content fits to the learner profile. Therefore, we need to model the learner and the course content. The matching process is between the style of the learner and the type of the content. The style of the learner is configured using learning style theories. In this study, we used 30 content types. These are as given in Table4.1.

Table 4.1: 30 Content Types used in the framework

Activity	Advance Organizer	Audio	Concept
Concept Map	Critique	Data	Definition
Diagram	Discussion	Example	Exercise
Experiment	Fact	Formula	Image
Index	Innovation	New Concept	Principle
Problem solving	Procedure	Question	HyperText
Slide	Syllabus	Table	Text
Theory	Video		

Different content types have been used in previous studies. The content can be a diagram, question, exercise, experiment, figure, graph, text, table, or slide which is retrieved from type of learning resources defined by IEEE LOM Metadata specification[18]. Howard et. al. used hyperText, slides, audio and video content types were used in the study [75]. Concept, example, activity are the content types referred in [76]. Diagram, fact, procedure, innovation, theory content types are mentioned as type of learning resources in [33]. The rest of the content types are derived from the descriptions provided by the learning style model[14].

According to the learner profile, we try to find the appropriate content using these content types.

4.1 Learner Profile

Learner modeling is crucial to provide adaptive instruction. Each learner requires an individualized student model. The better we model our learners, the more we know the personalization needs of them. To know the learner, we need to keep a variety of information about the learner, such as, learning styles, domain knowledge, progress, preferences, goals, interests, etc. In this study, as in[77], we model the learner according to three factors which are behavioral factors, knowledge factors, and personality factors.

4.1.1 Behavioral Factors

The behavioral factors involve the actions of the learner performed in the LMS module. The learner action information involves the following fields:

- identity of the action,
- identity of the learner,
- owner of the action (LMS or any other systems),
- name of the action (search, view lecture notes, login etc.),
- time to start action,
- time to end action
- description of the action
- number of occurrences of each 32 media types (In case of view content action, if the learner views a content with 3 charts and 1 concept map, than increase the count for these types)

4.1.2 Knowledge Factors

Our framework is based on the idea that the course content material must be kept away from the adaptive learning system. The adaptive learning system is supposed to apply a mechanism to select the appropriate content type resource to the learner. To achieve this, it is not necessary to keep the all information about the learning resources. The following fields are found sufficient to store the learner's knowledge levels on the course content.

- identity of the content

- identity of the content item (each course concept)
- type of the content item (one or more of 32 content types)
- the knowledge level of the learner (UNDERSTOOD, NOT UNDERSTOOD, MISUNDERSTOOD)
- exam results
- last modified date

4.1.3 Personality Factors

To keep variety of information about the learner, we use both the learning style information and the learning standards. There are several standards defined for modeling the learner. Two most popular standards are IMS LIP(Instructional Management Systems Learner Information Package)[78], IEEE PAPI(Public and Private Information)[79]. In this study, we chose IMS LIP as the learning standard. It is possible to convert IMS LIP to IEEE PAPI whenever needed. The IMS LIP specification addresses the interoperability of the Internet-based learner information systems with other systems that support the Internet learning environment. The IMS LIP involves the information such as biographic and demographic data relevant to learning; career and other objectives and aspirations; qualifications, certifications and licenses granted by recognized authorities; any learning-related activity(informal education, training, work experience, and military or civic service) in any state of completion; transcript; information describing hobbies and recreational activities; skills, knowledge, and abilities acquired in the cognitive, affective, and/or psychomotor domains; membership of professional organizations; the set of passwords, and security keys. It is possible to obtain more information about IMS LIP at http://www.imsglobal.org/profiles.

Although the standards cover lots of information, they are incapable of providing the personalization information[77]. Learning style is very important in adaptation. For this reason, IMS LIP does not provide this information. Therefore, we decided to use the personality factors including the learner's IMS LIP and the learning style information. Learners have different ways of perception, construction, and retention of knowledge. These differences which occur during the learning process are unique to each individual based on many factors like previous experiences, mental abilities, and personal characteristics. In order to provide adapted instruction, the learning styles must be concerned and instructors must ask *how can this learner achieve more?* instead of *why is this learner not high-achiever?*. There is different learning style models used in the literature. These are as follows:

- Dunn, Dunn and Price Model[29]
- Felder-Silverman Model[14]
- The Myers-Briggs Type Indicator[25]
- Kolb's Learning Style Model[28]
- Honey and Mumford's Typology of Learners[15]

In this study, the Felder-Silverman model was taken as the core learning style model which is one of the most widely used models. The Felder-Silverman model classifies students according to where they fit on a number of scales pertaining to the ways they receive and process information. Felder and Silverman provides definitions of learning styles which are described as follows[35]:

Active Active learners like to try things out and see how they work and like to work with others.

Reflective Reflective learners like to think things through first.

Sensing Sensors like to learn facts, use well established methods. They are practical and careful.

Intuitive Intuitors tend to work fast and be innovative and can often handle abstract and mathematical concepts well.

Visual Visual learners like diagrams, pictures, graphs and films.

Verbal Verbal learners get more out of words heard and written.

Sequential Sequential learners like to work in linear steps that follow logically.

Global Global learners like to jump in, absorb material nearly at random and then get the big picture.

The fields kept about personality factors in the framework as follows:

- IMS(Instructional Management System) Learner Information Package
  - qualifications, certifications and licenses granted by recognized authorities
  - learning-related activity (informal education, training, work experience, and military or civic service)
  - transcript
  - hobbies and recreational activities
  - skills, knowledge, and abilities acquired in the cognitive, affective, and/or psychomotor domains
  - membership of professional organizations
- Learning Style Information

  - Sensing
  - Intuitive
  - Active
  - Reflective
  - Visual
  - Verbal
  - Sequential
  - Global

- Additional Fields
  - Last update date of the style information
  - Last update date of the IMS LIP information

4.2 Course Content Profile

Similar to the learner profile, we construct a course content profile. Course content profile has the following fields:

- Identity of the course
- Number of occurrences of each 30 content types

As an example, suppose we are offering a Java Programming course. We have several lecture notes for the same course, each lecture note having different content types, namely course content summaries. Table4.2 gives different course content summaries.

Note that some of the content types might overlap. For example, a content object may involve a text content type explaining a fact. In this case, the content type counts are calculated separately.

Table 4.2: Samples Content Type Definitions of Content Resources

Contents	**Activity**	**Audio**	**Chart**	**Concept**	...	**Video**
I	4	7	8	5	...	2
II	2	1	6	3	...	9
III	14	4	12	9	...	5

4.3 Course Content Classification

In this study, we provide a method to classify the learner actions and update the learner style information. This method bases on the learning style dimensions. Since the learner actions are classifies according to using 32 content types, we need to specify the contents and the corresponding learning style dimensions in the same way. According to Felder[14],

- Sensors like facts, data, and experimentation; whereas intuitors prefer principles and theories. Sensors like solving problems by standard methods and dislike "surprises", intuitors like innovation and dislike repetition.

- Sensors are patient with detail but do not like complications; intuitors are bored by detail and welcome complications.

- Sensors are good at memorizing facts; intuitors are good at grasping new concepts.

- Intuitors are more comfortable with symbols than are sensors.

According the Felder-Silverman learning style, the properties of the visual and verbal learners can be summarized as that visual learners remember best what they see: pictures, diagrams, flow charts, time lines, films, and demonstrations. If something is simply said to them they will probably forget it. Auditory learners remember much of what they hear and more of what they hear and then say. They get a lot out of discussion, prefer verbal explanation to visual demonstration, and learn effectively by explaining things to others. Felder and Silverman[14] explain the active learners as experimentalists, the reflective learners as theoreticians. He states that active learners need to be active in order to learn, and reflective learners need to have an opportunity to think about the information being presented. It is obvious that when the students in a class are passive, then neither active learner not reflective learner can learn effectively. As explained in the learning style model, the active learners work well in groups. However, the reflective learners work better by themselves or with at most one other person.

Felder and Silverman[14] suggest to present material that emphasizes both practical problem solving (active) and fundamental understanding (reflective). It might be good to present lectures with occasional pauses for thought (reflective) and brief discussion or problem-solving activities (active).

An exceptionally effective technique for reaching active learners is to have students organize discussion groups in the forum and let them come up with collective answers to the questions posed by instructor. To support this, the AILS agents ask the learner to work with the other learners who have common goals which are kept in the IMS LIP standard information. Sequential learners may be strong in convergent thinking and analysis; global learners may be better at divergent thinking and synthesis. Sequential learners learn best when material is presented in a steady progression of complexity and difficulty; global learners sometimes do better by jumping directly to more complex and difficult material. The curricula, course syllabi, textbooks, the lecture notes prepared for the online education are generally designed as sequential, which is already suitable for sequential learners. In order to reach the global learners, it might be good to provide overall picture of the course or goal of a lesson before presenting the steps, doing as much as possible to establish the context and relevance of the subject matter and to relate it to the students' experience. Another way to support global learners is to explain their learning process to them. According to Felder and Silverman[14], while global are painfully aware of the drawbacks of their learning style, it is usually a revelation to them that they also enjoy advantages that their creativity and breadth of vision can be exceptionally valuable to future employers and to society. This background information guides us to define the learning resources and the corresponding learning style dimensions as given in Table4.3.

Table 4.3: The Learning Resources and the Learning Style Dimensions

Learning Resource Type	Dimension
DATA, DEFINITION, EXPERIMENT, FACT, PROCEDURE, CONCEPT	Sensing
FORMULA, THEORY, PRINCIPLE, INNOVATION, NEW CONCEPT	Intuitive
IMAGE, DIAGRAM, VIDEO, SLIDE, TABLE	Visual
AUDIO, TEXT	Verbal
EXAMPLE, EXERCISE, ACTIVITY, DISCUSSION, PROBLEM SOLVING	Active
QUESTION, CRITIQUE	Reflective
HYPERTEXT	Sequential
CONCEPT MAP, SYLLABUS, INDEX ADVANCE ORGANIZER	Global

Using this content type and learning style classification, we update the learner profile. The classification process is as follows: Let A=[a_1,a_2,a_3, a_n,a_m] denote the learner actions and suppose the learner profile is already updated considering the actions up to action a_n. Then, the classification and update of the learner profile starts with the action a_n and ends with a_m. The values of 30 dimensions of this action are classified into 8 dimensions using the table above. The learner profile is updated and the last action a_m is stored as a result.

4.4 Learner Profile

The learner profile is initialized according to the personality factors, behavior factors, and knowledge factors. For the personality factor, it is suggested to use questionnaires. Since the study is based on the Felder-Silverman learning style, there is already a well-known questionnaire used to categorize individual learning styles, called Felder-Solomon Index of Learning Style (ILS)[35]. This questionnaire is validated in[34].

Knowledge factors and the behaviors of the learner are initially set to their default values. They are updated as the learner interacts with the learning system.

Most of the current adaptive systems except iWeaver and MANIC assess the learning styles through psychometric questionnaires. The disadvantage of this approach is that the learners are classified into stereotypical groups and the assumptions about their learning styles are not updated during the following interaction with the system[33].

In this framework, the learner profile is updated for different cases. During the learning process, behavioral factors of the learner is updated according to the learner actions using the fields explained in section4.1.1.

Updating the learner style information requires a classification and evaluation of the learner actions, choices, and preferences. This results in making classification on the profile or even to learn the patterns and associations in the learner's profile. To learn the profile, one might use machine learning and data mining techniques on the profile data.

In this study, we provide a method to classify the learner actions and update the learner style information according to this classification.

Let $A = [a_1, a_2, a_3, a_n, a_m]$ denote the learner actions and suppose the learner profile is already updated considering the actions up to action a_n. Then, the classification and update of the learner profile starts with the action a_n and ends with a_m. The action's 32 dimension values are classified into 8 dimensions using the table given in the section4.1.1. Then, the learner profile is updated and the last action a_m is stored.

4.5 Learner-Course Content Matching

According to the framework, the adapted content is the content that is best matched with the learner profile. Therefore, we need to define a matching mechanism between the learner profile and the course content profile. There might be several approaches for calculating the matching rates of the course contents. In this study, we simply use Euclidian distance and find a matching score based on the normalized distance.

We keep the learner style information in eight dimensions, $x = [x_1, x_2, .., x_8]$. We have course content profile information classified into eight dimensions of learning style, $y = [y_1, y_2, .., y_8]$. We normalize the dimension values by substituting the maximum value of the dimension. Then, the resulting normalized vectors become:

$$x_n = [x_1/x_m,\ x_2/\ x_m,\ ..,\ x_8/\ x_m]$$
$$y_n = [y_1/y_m,\ y_2/\ y_m,\ ..,\ y_8/\ y_m]$$

where x_m is the maximum value in the x vector and y_m is the maximum value in the y vector.

The Euclidian distance between these two dimensions is computed as:

$$D(x, y) = \parallel x - y \parallel \tag{4.1}$$

$$= \sqrt{(x_{1n} - y_{1n})^2 + (x_{2n} - y_{2n})^2 + ... + (x_{8n} - y_{8n})^2} \tag{4.2}$$

The matching score is defined as:

$$S(x, y) = -D(x, y) \tag{4.3}$$

The $S(x, y)$ gives the matching score for the learner and the course content profiles. The matching score is calculated for each course content profile. The resulting scores are sorted and the course content with the highest score is accepted as the best fitted course content regarding the learner profile. After finding and sorting the scores, a filtering might be applied depending on the application choices. One might prefer to display the first three best matched contents or two, or may accept only the best one.

In fact, there might be many ways to calculate the matching scores between the learner and the content profiles. This is one of the areas opened to further research.

CHAPTER 5

MODA: A Multi-Agent Adaptive Learning System for any LMS

This chapter presents a multi-agent, called MODA, developed to provide adaptiveness in learning management systems (LMS). The adaptiveness provides uniquely identifying and monitoring the learner's learning process according to the learner's profile. This chapter covers the architecture of MODA and its agents, the protocol providing communication between MODA and LMS, and a sample application of MODA to an open source learning management system, OLAT.

5.1 Components of MODA

The aim of MODA is to integrate adaptive behavior into online learning management systems. The main components of the system are as follows:

Learner Modeling MODA provides online learners a personalized learning experience. It maintains a learner profile. The detail description of the profile is provided in the following section.

Adapted Content MODA chooses the most appropriate course content by matching the learner and course content using a strategy explained in section 4.5.When the learner performs a *show content* or *search* actions, the content or search results adapted to the learner is presented to the learner.

Integration with LMS The system can be integrated into any LMS. In order to provide this modularity, we provide an LMS-MODA communication protocol. Any LMS using this protocol can communicate with MODA and have the adaptive features.

5.2 The Pedagogical Architecture

The effectiveness of the adaptation in online learning environments is higly related to the coverage of the adaptation strategy. The better the match between the learner and the instruction is, the higher the adaptation is. We applied the conceptual framework developed in this study (See Chapter4) which defines the learner profile and the course content profile. The framework takes its background from learning styles and learning standards. The adaptation strategy in the framework is to find the best match between the learner and the instruction set.

5.3 The Multi-Agent System Architecture

Since MODA was developed as a multi-agent system, we used an agent-oriented software engineering modeling technique in order to model the system. The existing software development techniques (for example, object-oriented analysis and design) are not suitable for multi-agents systems' analysis and design tasks, because there is a fundamental mismatch between the concepts used by object-oriented developers and the agent-oriented view. Thus, we chose Gaia[20] as the agent-oriented software engineering technique. This methodology is an attempt to define a complete and general methodology that is specially tailored to the analysis and design of multi-agent systems (MASs). In Gaia, analysis and design can be thought as a process of developing increasingly detailed models of the system to be constructed[80]. The models of Gaia methodology is given in Figure5.1. The architecture of AILS was modeled with these Gaia models. In this section, these models are given in order to understand AILS and its environments.

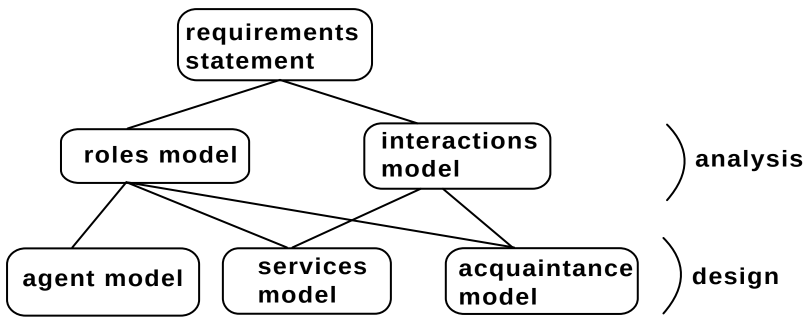


Figure 5.1: The Gaia Models

MODA was designed to work with an LMS. We have three main modules: LMS , MODA and LMS-MODA interface module (See Figure5.2). LMS can be any LMS providing online learning services to learners. MODA is the multi-agent system. It has several agents to perform the adaptive services required by LMS. LMS-MODA interface is the communication platform of these two separate modules. We developed a socket-based communication protocol. MODA can communicate with any LMS if LMS sends the data packets in the same format defined in the protocol.

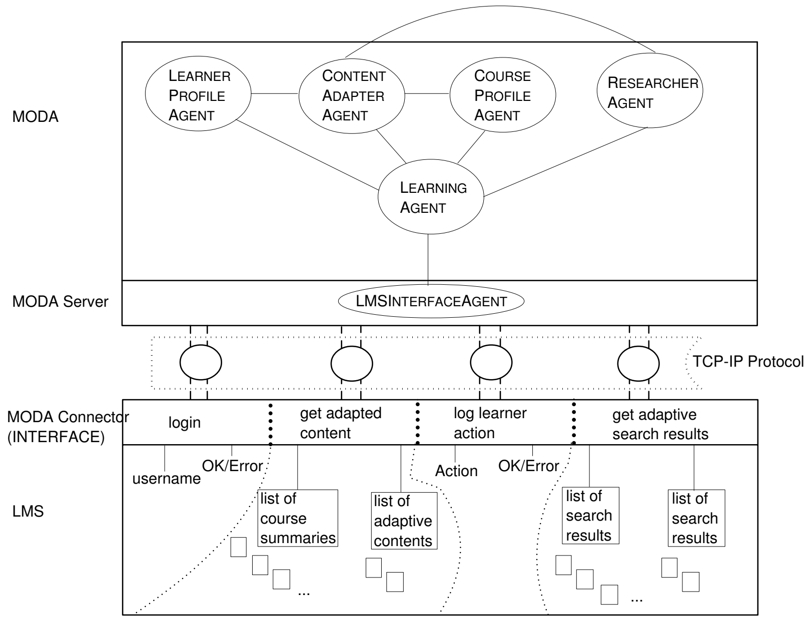


Figure 5.2: The Architecture of MODA

5.3.1 The MODA Agents

The system has seven learning agents: LMSInterfaceAgent, LearningAgent, ContentAdapterAgent, CourseProfileAgent, LearnerProfileAgent, ResearcherAgent and AgentManager. The descriptions and roles of each agent are as follows:

- LMSInterfaceAgent is the communication party with the LMS. It behaves as the MODA server. It receives MODA request from the LMS part and provides request to the LearningAgent. It receives results from the LearningAgent and provides MODA Respond message to the LMS.

- LearningAgent is the central agent which is responsible for management of the other agents.

- ContentAdapterAgent is responsible for finding the most appropriate content for the learner using the learner profile. This agent communicates with the LearnerProfileAgent to receive the learner profile information. After classification and matching, it sorts and filters the course contents. It communicates with the LearningAgent to send adapted course contents back. This agent also interacts with the ResearcherAgent to provide adapting search results.

- CourseProfileAgent initializes and updates the learner profile. The agent deals with the course profile classification of the course content types. It provides course profile information requested by the other agents.

- LearnerProfileAgent initializes and updates the learner profile. This agent updates the learner profile using learner actions. It provides the learner profile information requested by the other agents.

- ResearcherAgent receives search results, communicates with ContentAdapterAgent and receives the adapted content. This agent sends adapted search results back to LearningAgent.

The agents in MODA were developed as JADE agents (See Figure 5.3). JADE (Java Agent Development Framework)[19] is a software development framework aimed at developing multi-agent systems and applications conforming to FIPA(The Foundation for Intelligent Physical Agents) standards for intelligent agents. For more information about FIPA, one can refer to www.fipa.org. JADE is one of the most commonly used agent middleware and it has well-structured agent management mechanisms providing a runtime environment, a library of classes that programmers can use, and a suite of graphical tools that allows administrating and monitoring the activity of running agents.

Figure5.4 depicts the agent types used in the MODA, and the agent instances realize these agent types at run-time. For each online learner,

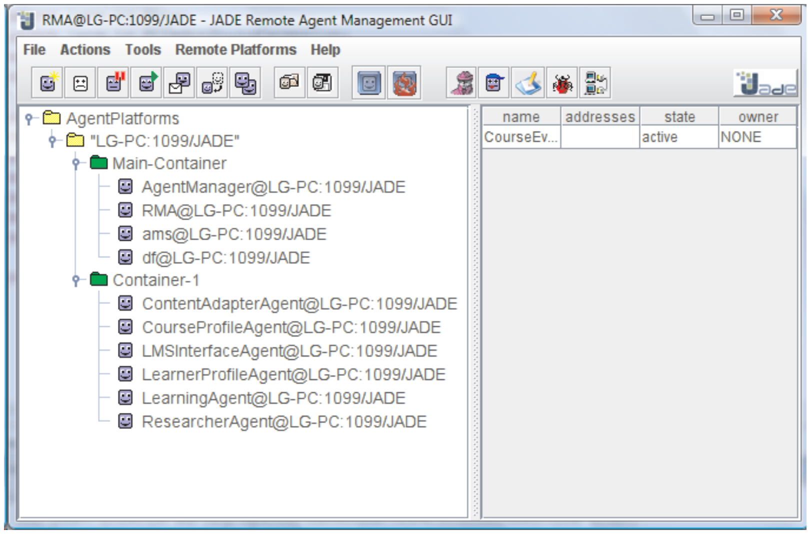


Figure 5.3: The JADE Environment with MODA Agents

each type of agent is instantiated once. In other words, there are as many agents as the number of online learners in the LMS at runtime.

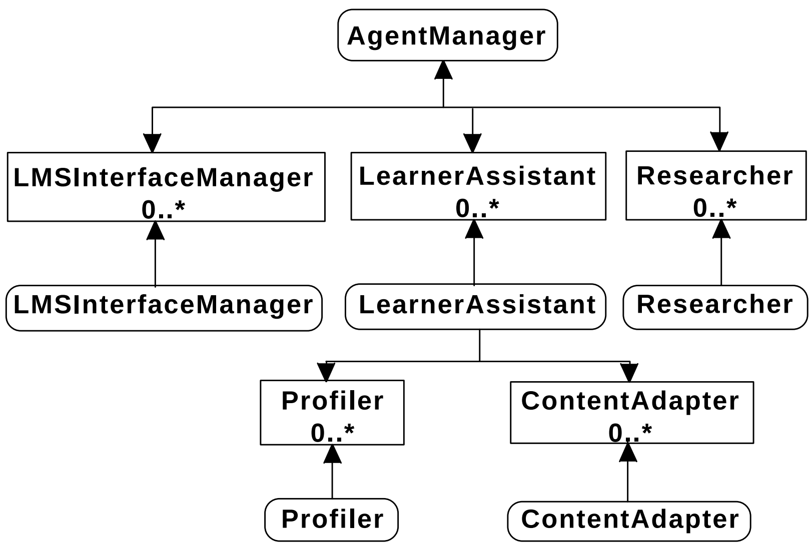


Figure 5.4: The Agent Model of MODA Agents

5.3.1.1 The Agent's Roles

The roles model identifies the basic skills required by the system. An agent can play one or more roles, to accomplish which agents typically need to interact with each other to exchange knowledge and coordinate their activities[20].

The AILS Agents has seven major roles. The role models of each agent role are given in the Figure 5.5 - 5.11.

Role:	LMSInterfaceAgent(LIM)
Description:	This role provides establishing a communication platform between MODA nad LMS. It receives actions from LMS module and then activates the corresponding function in AILS module communicating wiht other roles(agents) in AILS. It also provides AILS to activate some functions in LMS.
Protocols & Actvities:	SetLMSEvent, SetMODAEvent, ActivateMODAFunctions, ActivateLMSFunction
Permissions:	reads *acquaiantances* *// acquaiantances data structure*
Responsibilities:	**Liveness:** LMSINTERFACEAGENT = (SetLMSEvent.ActivateMODAFunction) $|^{\omega}$ (SetMODAEvent.ActivateLMSFunction)$^{\omega}$ **Safety:** A successful connection with LMS is established

Figure 5.5: The Role Model of LMSInterfaceAgent

Role:	LearnerProfileAgent(LPA)
Description:	This agent is responsible for initializing and update of the learner profile. It communicates with the other agents to provide the learner profile information.
Protocols & Actvities:	GetUser, InitializeProfile, ActivateProfile,UpdateProfile,GetProfileRequest,SendProfile
Permissions:	reads *acquaiantances* *// acquaiantances data structure* *reads learner profile* *write to learner profile*
Responsibilities:	**Liveness:** LEARNERAPROFILEAGENT **= ((**GetUser.InitializeProfile) || (ActivateProfile)|| (UpdateProfile) || (GetProfileRequest.SendProfile)) **Safety:** A successful connection with the database and LMS is established

Figure 5.6: The Role Model of LearnerProfileAgent

5.3.1.2 The Agent's Services

The aim of the Gaia services model is to identify the services associated with each agent role, and to specify the main properties of these services. A service is a function of the agent[20]. The services provided by the AILS agents are given in Figure 5.12. Each service of AILS identifies the

Role:	ContentProfileAgent(LPA)
Description:	This agent is responsible for classification and preparation of the course profile for the given course contents
Protocols & Actvities:	GetProfileRequest, PerformClassification, SendProfile
Permissions:	reads *acquaiantances // acquaiantances data structure* *reads contents* *write to content profiles*
Responsibilities:	
	Liveness: CONTENTPROFILEAGENT = (GetProfileRequest.PerformClassification.SendProfile)
	Safety: A successful connection with the database and LMS is established

Figure 5.7: The Role Model of COURSEPROFILEAGENT

Role:	LearningAgent(LA)
Description:	This agent is responsible for monitoring and managing the other MODA agents. It communicates with LMSInterfaceAgent and receives the learner actions and it provides the necessary data to the appropriate agent to perform the adaptation depending on the user request.
Protocols & Actvities:	GetLearnerRequest,PerformAdaption RespondToLearner
Permissions:	reads *acquaiantances // acquaiantances data structure*
Responsibilities:	
	Liveness: w LEARNERAGENT = (GetLearnerRequest.PerfomAdaptation.RespondToLearner)
	Safety: A successful connection with the database and LMS is established

Figure 5.8: The Role Model of LEARNINGAGENT

inputs, outputs, pre-conditions, and post-conditions of each service. These services are derived from the protocols, activities, responsibilities and the liveness properties of a role models described in the section5.3.1.1.

Role: ContentAdapterAgent(CAA)

Description: This is the most complex role serving to online learners in MODA. It receives the learning resources and the learner username. It communicates with LearnerProfileAgent to receive learner profile; CourseProfileAgent to receive the course content profiles. It calculates a mathicng score between the learner profile and the course profiles. It filters and sorts the resources according to the best match.

Protocols & Actvities: ReceiveLearningResources,ReceiveUsername,RequestContentProfile, RequestLearnerProfile, PerformAdaptation, CalculateScores, Sort, Filter, FindBestMatch,SendAdaptedResources

Permissions: reads *content* //lecture notes of the course which are in LOs
updates *content*
reads *acquaiantances* *// acquaiantances data structure*

Responsibilities:

Liveness:

CONTENTADAPTERAGENT = (ReceiveLearningResources.ReceiveUsername.
RequestContentProfile.RequestLearnerProfile.
PerformAdaptation.SendAdaptedResources)

PERFORMADAPTATION = CalculateScores.Sort.Filter.FindBestMatch

Safety:

A successful connection with the database and LMS is established

Figure 5.9: The Role Model of ContentAdapterAgent

Role: ResearcherAgent(RS)

Description: This role provides adapting search results obtained from LMS. It communicates with ContentAdapter and LearningAgent.

Protocols & Actvities: ReceiveSearchRequest, AdaptSearchResults, UpdateProfile, GetSearchResults, SendResults

Permissions: reads *learningObjects* // *learning objects in the repository*
reads *acquaiantances* *// acquaiantances data structure*
reads *profileData* *// profile data of the user*
updates *profileData* *// update data about keywords searched*

Responsibilities:

Liveness:

RESEARCHERAGENT = ReceiveSearchRequest.
AdaptSearchResults.GetSearchResults.
(UpdateProfile || SendResults)

Safety:

A successful connection with the database and other MODA is established

Figure 5.10: The Role Model of ResearcherAgent

Role:	AgentManager(AGM)
Description:	This role serves as a personal assistant of the multi agent system, initializing the system parameters, launching new agents when necessary and monitoring the agents' operation. Agents may register their services with the Agent and query the Agent to find out what services are offered by other agents. It also gets acquainted with specifc agents.
Protocols & Actvities:	RegisterAgent, QueryAgent, SaveNewAcquaintance, IntroduceNewAgent, Monitor
Permissions:	creates *acquaiantances* *// acquaiantances data structure* reads *acquaiantances* updates *acquaiantances*

Responsibilities:

Liveness:

AGENTMANAGER = (GetAcquaintance. (MeetSomeone)^{w}) \|\| (Monitor)^{w}
GETACQUAINTANCE = RegisterAgent.QueryAgent.[IntroduceNewAgent]
MEETSOMEONE = IntroduceNewAgent.SaveNewAcquaintance

Safety:
true

Figure 5.11: The Role Model of AGENTMANAGER

Service	Adapt Content
Inputs	-
Outputs	The description of the event
Pre-condition	A LMSInterfaceAgent, LearningAgent, CourseAdapterAgent, ContentProfileAgent, and LearnerProfileAgent are instantiated and associated with the MODA module.
Post-condition	-

Service	Search Learning Object
Inputs	The searching criteria
Outputs	A list of learning objects
Pre-condition	A Researcher agent is instantiated and associated with the learner
Post-condition	Learner views the learning objects

Service	LMS Integration
Inputs	The course events and learner logs
Outputs	Learner's actions in LMS
Pre-condition	A LMSInterface agent is instantiated and associated with the AILS
Post-condition	The communication between LMS and AILS is established

Service	Learning by Profile
Inputs	The learner profile
Outputs	Learner's actions in LMS
Pre-condition	A LMSInterface agent is instantiated and associated with the AILS
Post-condition	The communication between LMS and AILS is established

Figure 5.12: The Services Model of MODA

5.3.1.3 The Acquantiances of Agents

The acquaintance models simply define the communication links that exist between agent types[20]. This model does not define the messages sent or received. It simply indicates that communication pathways exist. Figure5.13 shows the communication and dependency between the agents.

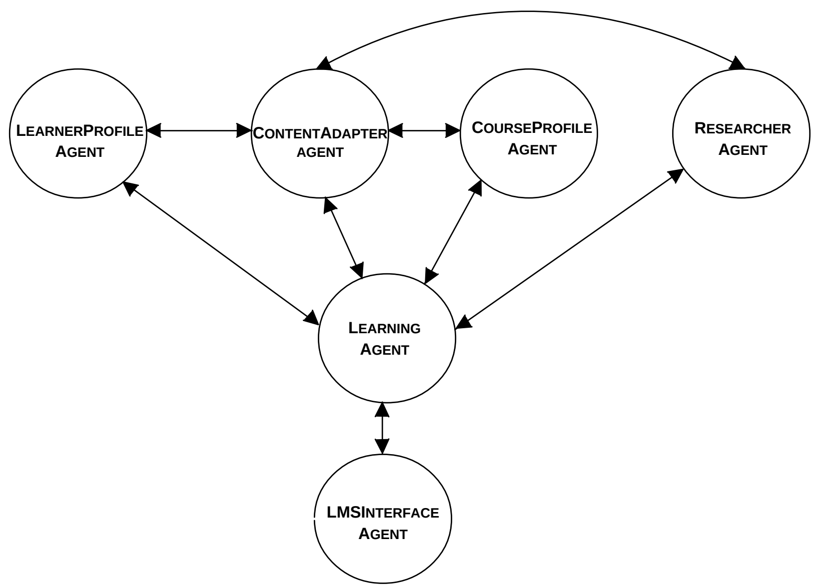


Figure 5.13: The Acquaintance Model of MODA Agents

5.3.2 The Agents' Behavior

MODA agents work cooperatively to perform the operations such as login, search keywords, and view lecture notes. For the sake of simplicity, each operation is explained as scenarios such as *login*, *show content* and *search* below.

5.3.2.1 Login

1. User enters username and password
2. LMS invokes LMSInterfaceAgent about the login and provides the user information to MODA

3. LMSInterfaceAgent sends login action to LearningAgent

4. LearningAgent receives new user action

5. LearningAgent sends user information to LearnerProfileAgent

6. LearnerProfileAgent receives user information

7. LearnerProfileAgent activate user profile if s/he exists in profile repository

8. LearnerProfileAgent initializes the profile if the user is a new user

9. LearnerProfileAgent send status of activation or initialization as SUCCESS or ERROR message to LMSInterfaceAgent.

10. LMSInterfaceAgent receives the status and sends to LMS

Figure5.14 depicts the agent communication for the login scenario.

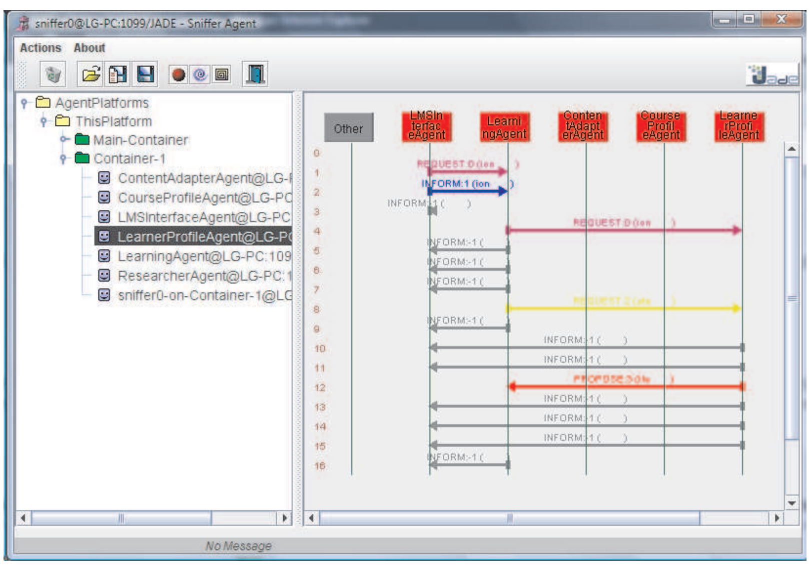


Figure 5.14: The Sniffing Login Behavior

5.3.2.2 Show Content

1. Learner clicks "Show Content"
2. LMS performs content searching for the course
3. LMS provides all available course content materials to LMSInterfaceAgent
4. LMSInterfaceAgent receives all course content information
5. LMSInterfaceAgent sends all available course content information to LearningAgent
6. LearningAgent receives and sends all available course content information to ContentAdapterAgent to adapt the content
7. ContentAdapterAgent receives course contents
8. ContentAdapterAgent requests learner profile information from LearnerProfileAgent
9. LearnerProfileAgent receives request for learner profile
10. LearnerProfileAgent prepares the profile
11. LearnerProfileAgent sends the learner profile information to ContentAdapterAgent
12. ContentAdapterAgent requests course content profiles of each course content from CourseProfileAgent
13. CourseProfileAgent receives available course contents
14. CourseProfileAgent classifies the course contents and prepares course profiles

15. CourseProfileAgent sends course profiles to ContentAdapterAgent
16. ContentAdapterAgent receives the learner profile
17. ContentAdapterAgent receives the course content profiles
18. ContentAdapterAgent calculates the matching scores between the learner profile and each course content profiles
19. ContentAdapterAgent sorts and filters the matching scores
20. ContentAdapterAgent sends best matched course content information to LearningAgent
21. LearningAgent receives resulting adapted course content list
22. LearningAgent sends adapted course content list to LMSInterfaceAgent
23. LMSInterfaceAgent receives course content list
24. LMSInterfaceAgent sends course content list to LMS
25. LMS receives adapted course content list and displays the content in adapted way.

Figure5.15 shows the sequence diagram for agent communication while performing the viewing lecture notes scenario.

5.3.2.3 Search

The agent communication for this scenario is given in Figure5.16.

1. Learner searches for a keyword in LMS
2. LMS performs searching for keyword

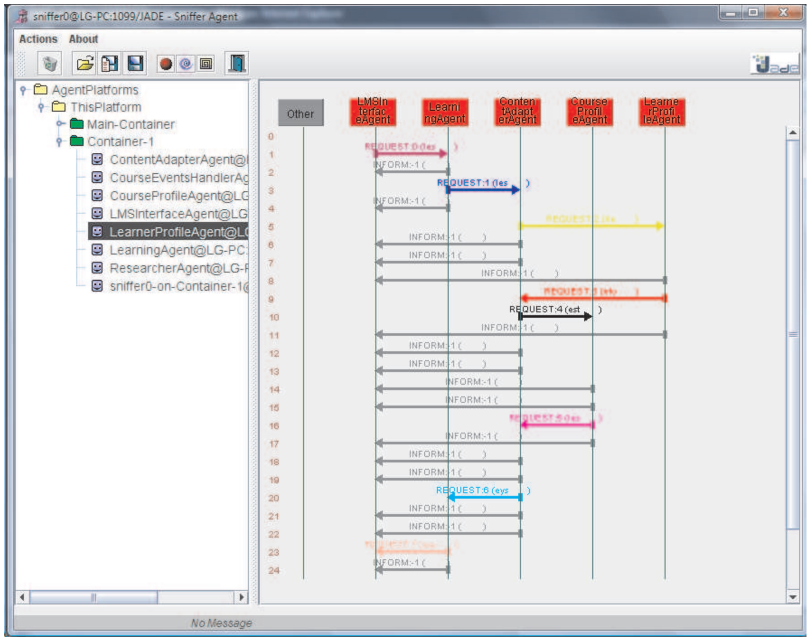


Figure 5.15: The Sniffing Show Content Behavior

3. LMS provides all available search results to LMSInterfaceAgent

4. LMSInterfaceAgent receives all search results

5. LMSInterfaceAgent sends all available search results information to LearningAgent

6. LearningAgent receives and sends all available search results information to ResearcherAgent to adapt the search results

7. ResearcherAgent receives the search results

8. ResearcherAgent sends search results to ContentAdapterAgent to perform adaptation

9. ContentAdapterAgent receives all available search results

10. ContentAdapterAgent requests learner profile information from LearnerProfileAgent

11. LearnerProfileAgent receives request for learner profile

12. LearnerProfileAgent prepares the profile

13. LearnerProfileAgent sends the learner profile information to ContentAdapterAgent

14. ContentAdapterAgent requests content profiles of each search result from CourseProfileAgent

15. CourseProfileAgent receives available search results

16. CourseProfileAgent classifies the contents and prepares search result content profiles

17. CourseProfileAgent sends search result content profiles to ContentAdapterAagent

18. ContentAdapterAgent receives the learner profile

19. ContentAdapterAgent receives the content profiles

20. ContentAdapterAgent calculates the matching scores between the learner profile and each content profiles

21. ContentAdapterAgent sorts and filters the matching scores

22. ContentAdapterAgent sends best matched content information to ResearcherAgent

23. ResearcherAgent receives resulting adapted content list

24. ResearcherAgent sends adapted content list to LearningAgent

25. LearningAgent receives resulting adapted search results list

26. LEARNINGAGENT sends adapted search result list to LMSINTERFACEAGENT

27. LMSINTERFACEAGENT receives search result list

28. LMSINTERFACEAGENT sends search results list to LMS

29. LMS receives adapted content list and displays the search results in adapted way.

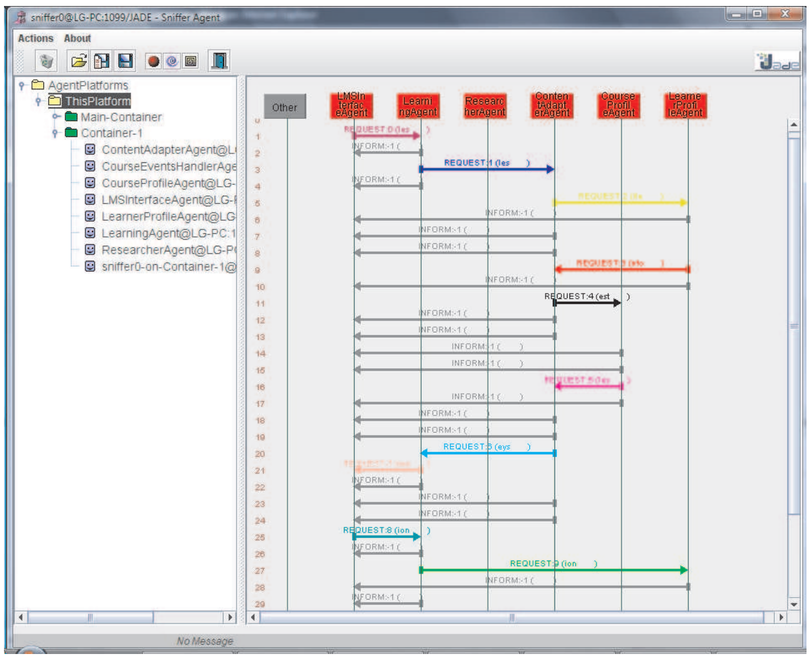


Figure 5.16: The Sniffing Search Behavior

5.3.3 The Ontologies

Messages exchanged by JADE agents have a format specified by the agent communication language (ACL) language defined by FIPA (http://www.fipa.org) international standard for agent interoperability. This format comprises a number of fields such as the sender, the list of receivers, the content, content language, the ontology etc[19]. The jade ontology describes the vocabulary of symbols used in the content and their meaning. Both the sender and the receiver must ascribe the same meaning to symbols for the communication to be effective.

In agent's communication, both the sender and the receiver must ascribe the same meaning to symbols for the communication to be effective. To achieve this, JADE defines ontology as the vocabulary of the symbols used in the content and their meaning. The JADE ontology is a set of schemas defining the structure of the predicates, agent actions and concepts (basically their names and their slots) that are pertinent to that domain. As an ontology is basically a collection of schemas that typically does not evolve during an agent lifetime.

In MODA, six jade ontologies are developed in order to meet the requirements of the system. The ontologies are LearnerProfileOntology (See Figure 5.17), LearningContentOntology (See Figure 5.18), ActionHistory-Ontology (See Figure 5.19), LMSUserOntology (See Figure 5.20), Seach-CaseOntology (See Figure 5.21), and AgentMessageOntology (See Figure 5.22).

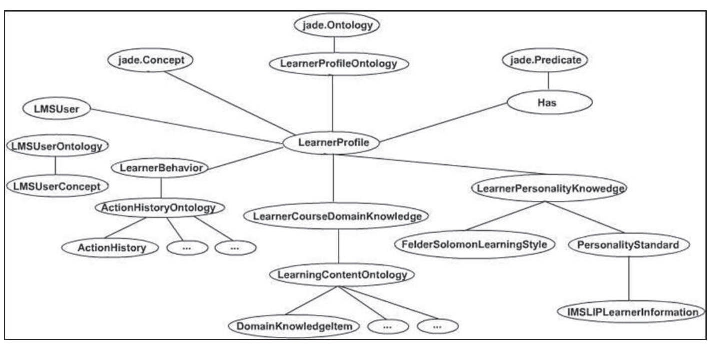


Figure 5.17: The Jade Ontology Map for the Learner Profile

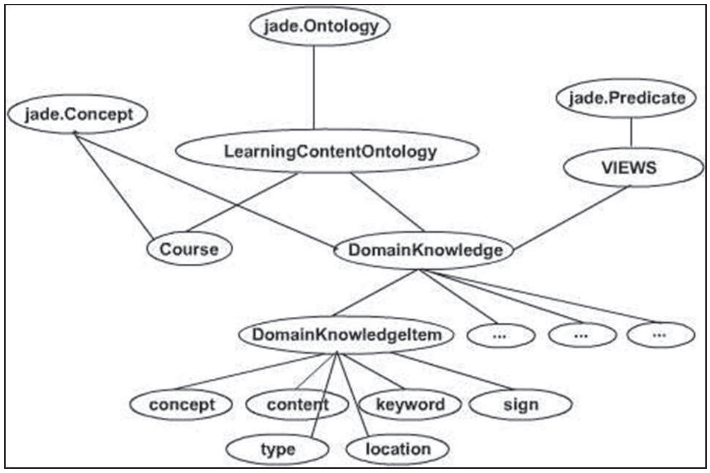


Figure 5.18: The Jade Ontology Map for the Learning Content

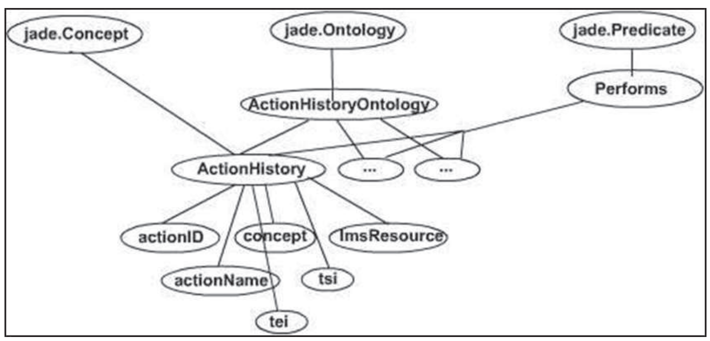


Figure 5.19: The Jade Ontology Map for the Learner Actions

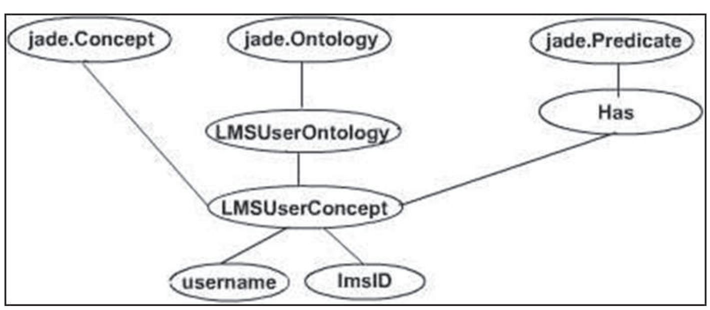


Figure 5.20: The Jade Ontology Map for the LMS User (learner)

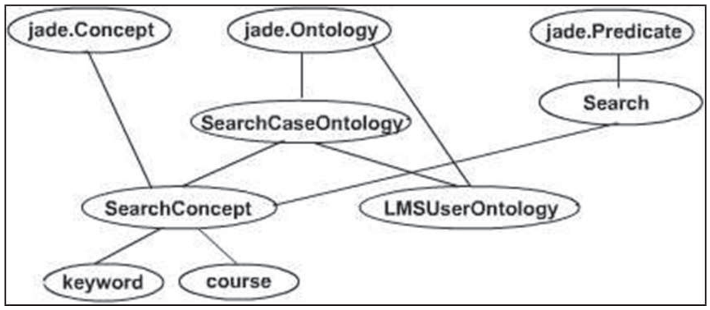


Figure 5.21: The Jade Ontology Map for the Search Case

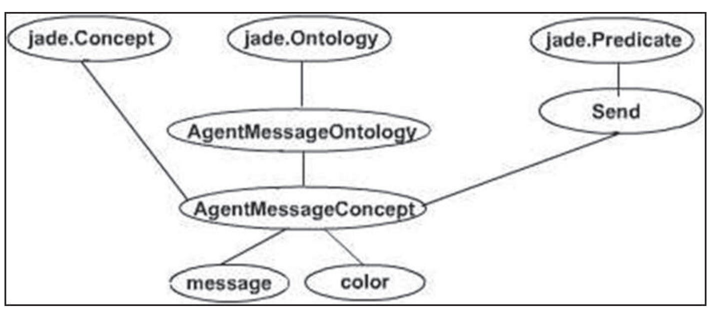


Figure 5.22: The Jade Ontology Map for the Agent Messages

5.4 LMS-MODA Communication Protocol

In the study, it is aimed that any LMS can make use of the adaptive learning system, MODA. In order to achieve this modularity, we provide a protocol for communication between LMS and MODA. Any LMS providing necessary information with the required format becomes an adaptive learning management system, when it establishes a communication with MODA. The protocol requires LMS and MODA to read/write the necessary information to TCP sockets. Since the communication occurs in the sockets, we define the data formats exchanged between the systems during either requesting data or responding a request. In MODA, one of the agents, - LMSInterfaceAgent-, serves as a server that receives the requests from the LMS and provides the responses back. In the following sections, we provide the communication scenarios, request and response data types, and the structure of each data type used in the communication between LMS and MODA.

5.4.1 Communication Scenarios between LMS and MODA

5.4.1.1 Login

When the user logs into the LMS, LMS notifies MODA.

5.4.1.2 Get Adapted Content

When the learner tries to access the content of a course, LMS sends course summary information of all available alternative contents to MODA. Then MODA compares the alternatives with the learner profile, sorts and filters the list according to the match scores, and then the sorted and filtered list is sent back to the LMS. Thereafter, LMS displays the first content in the list to the learner. The other contents in the list (if any) are displayed to the learner as available appropriate alternatives in a sidebar, or a similar menu list.

5.4.1.3 Log Learner Action

When a learner pays attention to a learning object in a course content, this is sent to the MODA. Then, MODA uses the history (log) of these actions to update the profile of the user. *Pays attention* has to be defined formally. This can be captured by learner's mouse clicks on a learning object or by understanding whether a learning objet is shown (visible) on the screen. This will be limited with the capabilities of the learning management system. The learner clicks on the syllabus and search tool are sent as learner actions to be logged. These actions will be used to update the learner's action history data on the fields *Critique* for search action and *Syllabus* for syllabus actions.

5.4.1.4 Search

When a learner performs a search action using the tools of LMS, LMS compiles the search results as a *list of course content* and sends this list to MODA. MODA compares this list with the learner's profile, and performs sorting and filtering. Then, the filtered and sorted list is sent back to the LMS. Finally, LMS displays the adapted search results to the learner.

The request and response messages exchanged between LMS and MODA are depicted in Figure5.23.

5.4.2 Communication between LMS and MODA

5.4.2.1 Data Packets

Communication is performed through data packets. A packet can be either a request or a response packet. The packet structure of request and response are provided in the protocol. Figure5.24 and Figure5.25 show the

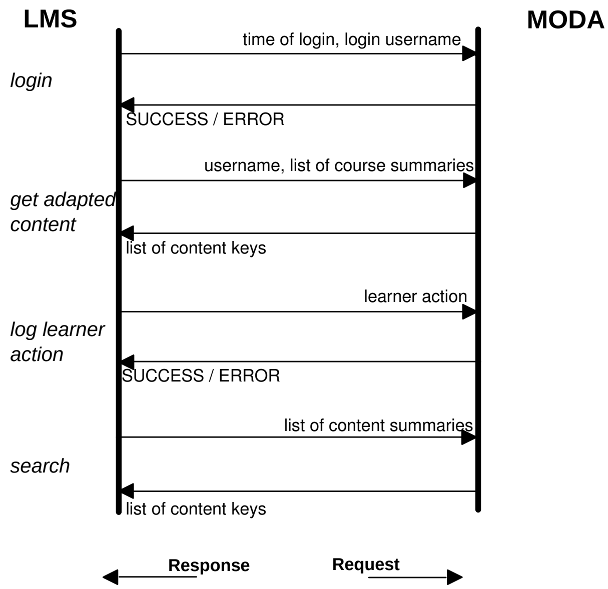


Figure 5.23: Request and Response Data in Communication Scenarios between LMS and MODA

structure of the request and response packets, respectively.

offset	Description	Length
0	Request Time	64 bits long miliseconds
8	Request Type	32 bit int
12	User name	20x8 bit ASCII ('\0' terminates)
32	Data Length	32 bit int
36	RESERVED	28 bytes
64	Data File Content	header #32 bytes. Data file formats are defined for each Request Type separately

Figure 5.24: Request Packet Structure

5.4.2.2 Data Files

The protocol defines the data files exchanged during the communications. There are four main data files defined in the protocol: NULL, COURSE-CONTENT-KEY-LIST, LOGIN, and GET-ADAPTIVE-CONTENT.

offset	Description	Length
0	Response Time	64 bits long miliseconds
8	Responded Request Time	64 bit long miliseconds
16	Responded Request Type	32 bit int
20	Responded Request user name	20x8 bit ASCII
40	Response Type	32 bit int
44	Data Length	32 bit int
48	RESERVED	16 bytes
64	Data File Content	header #48 bytes. Data file formats are defined for each Response Type separately

Figure 5.25: Response Packet Structure

NULL It is a special data file with zero length. It is used in responses, as necessary.

COURSE-CONTENT-KEY-LIST Keys are stored with their lengths, one after another. The byte length of the total list is given in the header as data file length (Figure5.26).

	offset	Description	Length
repeats	0	lengthOfNextKeyString	32 bit int
	4	next key content	ASCII
	4+length of previous	length of next	
	...	...	...

Figure 5.26: The Data File Structure for Course Content Key List

LOGIN Data file length is zero.

GET-ADAPTIVE-CONTENT The file structure is defined as in Figure5.27.

offset	Description	Length
0	Course Content Count	32 bit int
4	Course Content Key	20 x 8 ASCII
24	value for content type 0	32 bit int
28	value for content type 1	32 bit int
32	value for content type 2	32 bit int
36	value for content type 3	32 bit int
40	value for content type 4	32 bit int
44	value for content type 5	32 bit int
48	value for content type 6	32 bit int
52	value for content type 7	32 bit int
56	value for content type 8	32 bit int
60	value for content type 9	32 bit int
64	value for content type 10	32 bit int
68	value for content type 11	32 bit int
72	value for content type 12	32 bit int
76	value for content type 13	32 bit int
80	value for content type 14	32 bit int
84	value for content type 15	32 bit int
88	value for content type 16	32 bit int
92	value for content type 17	32 bit int
96	value for content type 18	32 bit int
100	value for content type 19	32 bit int
104	value for content type 20	32 bit int
108	value for content type 21	32 bit int
112	value for content type 22	32 bit int
116	value for content type 23	32 bit int
120	value for content type 24	32 bit int
124	value for content type 25	32 bit int
128	value for content type 26	32 bit int
132	value for content type 27	32 bit int
136	value for content type 28	32 bit int
140	value for content type 29	32 bit int
144	value for content type 20	32 bit int
148	value for content type 31	32 bit int

Figure 5.27: The Data File Structure for Course Content List

5.5 An Example Session

MODA has been integrated to an open source learning management system, OLAT[21]. You can obtain more information at http://www.olat.org/website/en/html/index.html. A session includes the following actions:

- login
- show content
- search keyword
- log learner action

We created a sample course in MODA and we provided different type of contents for the same course content item. We tested the system by entering as learners having different learner styles. The system gave the most suitable course content items for that type of learner. During this test, we had no problems with the communication protocol.

Sample screen for login page of OLAT is given in Figure 5.28.

Figure 5.28: Login Screen of OLAT

Figure 5.29 shows the welcome page of OLAT. We integrated MODA into OLAT and we created a OLAT tool in order to display the information about MODA to the user.

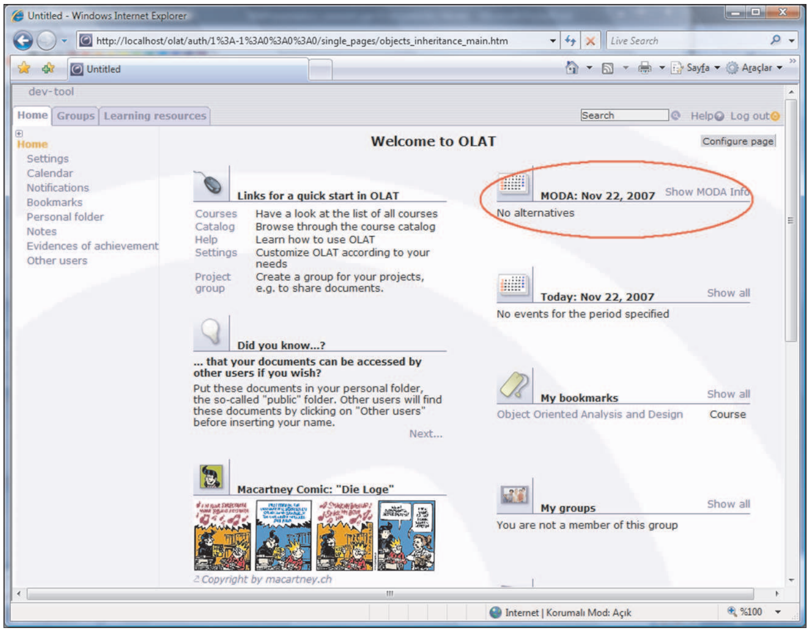


Figure 5.29: Welcome Page of OLAT

The learners access to the course content through Learning Resources tool and clicks *Show Content* link as in Figure 5.30.

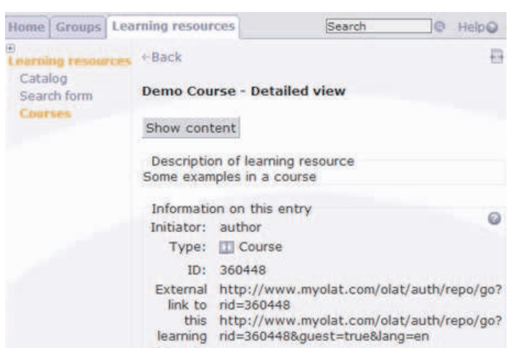


Figure 5.30: Show Content tool of OLAT

When the learner clicks *Show Content*, OLAT displays all available content resources to the learner as shown in Figure5.31 before adaptation.

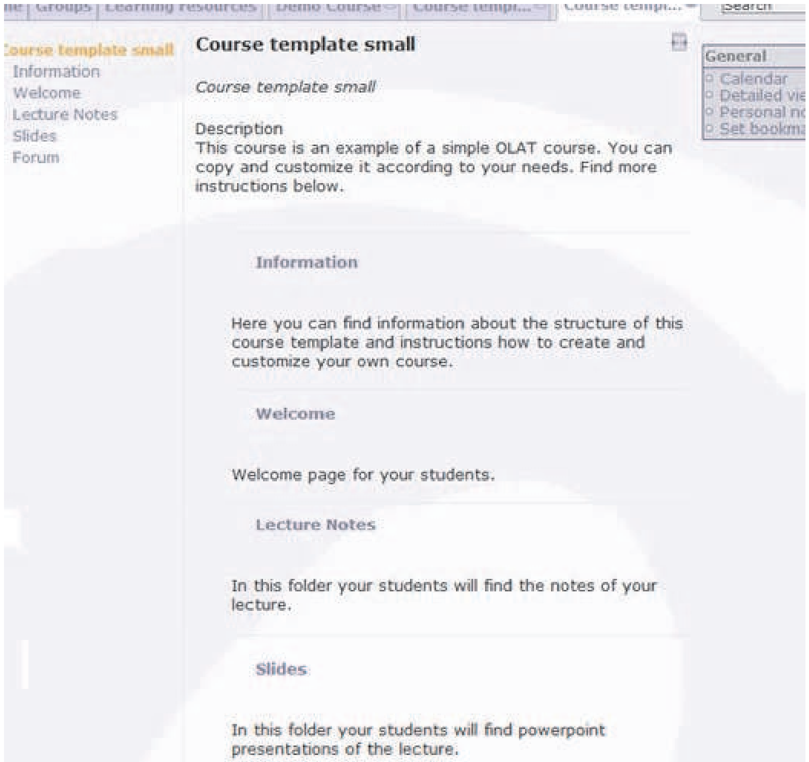


Figure 5.31: Show Content tool of OLAT before adaptation

Figure 5.32 gives the sample screen for the adaptive content. It can be easily shown in Figures5.31, the number of all available resources for the learner is six. However, when OLAT becomes an adaptive learning system, it adapts the content and shows the appropriate content to the learner. In this example, it founds only two of the content types suitable for the learner.

Similar to course contents, MODA adapts the search results. Figure 5.33 shows an example for search action in OLAT with MODA.

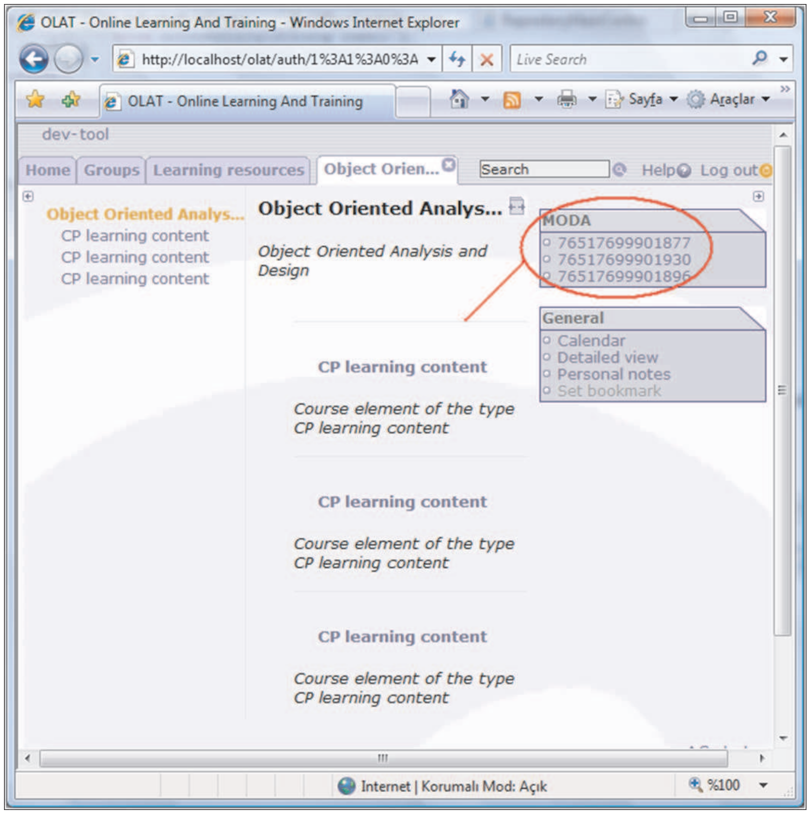


Figure 5.32: Show Adapted Content in OLAT integrated with MODA

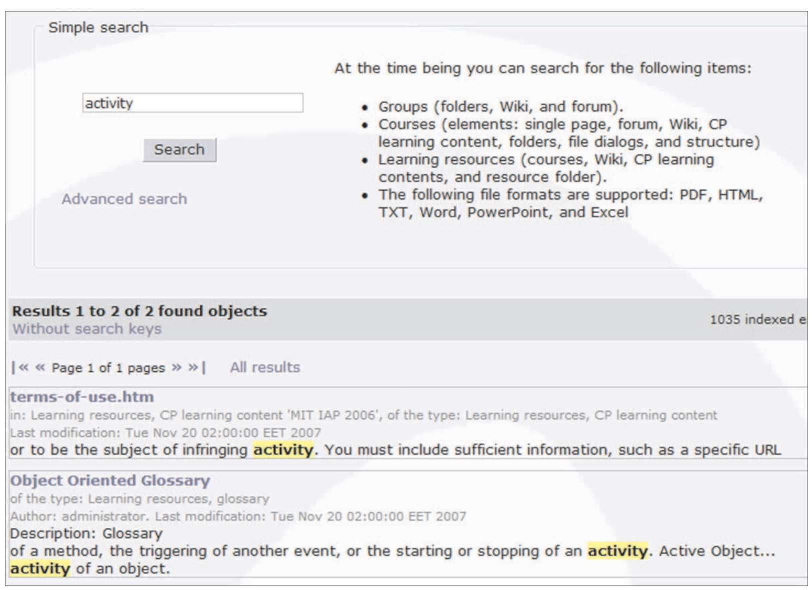


Figure 5.33: Search action in OLAT integrated with MODA

5.6 Evaluation of MODA

In the scope of this thesis, we successfully integrated MODA into a learning management System, namely OLAT. However we did not evaluate the system in a course to see the pedagogical effects of the system on learners. We have been in contact with OLAT users and developers. We are planning to work collaboratively with OLAT community and use MODA in the online courses given through OLAT.

Basically, the adaptive systems performs two functions such as acquiring data about the user, which are used to infer abstract characteristics, and deciding how to adapt based on these characteristics, which has impact on both the system behavior and the user behavior. As a further study, an evaluation study of MODA should at least include the following evaluation issues:

- Evaluation of reliability and external validity of input data acquisition
- Appropriateness of adaptation decisions in MODA
- Change of system behavior when MODA adapts
- Change of user behavior when MODA adapts

CHAPTER 6

Conclusion and Further Research

The adaptiveness provides uniquely identifying and monitoring the learner's learning process according to the learner's profile. It is a crucial issue in todays online learning environments. In service-job-training especially necessitates to identify learning needs and customize solutions that foster successful learning and performance, with or without an instructor to supplement instruction.

Moreover, there is a need for providing education to students in different places around the world where no teacher is available for face-to-face assistance in most cases. To support teaching and to facilitate learning, learning management systems must provide individualized help just as a human tutor would.

The intelligence of a learning management system is largely attributed to its ability to adapt to specific student needs during the learning process. Online learning environments have been used by a much wider variety of students. Each student may have different backgrounds, learning styles, individual preferences, and knowledge levels. This raises the need for adaptiveness of learning environments. The learning systems must be flexible to be be suitable for any particular kind of students.

In this study, we propose a conceptual framework for supporting adaptiveness in online learning environments. We claim that this framework will help both instructional designers and software developers to design and develop adaptive learning systems.

Our framework is based on the idea that the adaptiveness is the best matching between the learner profile and the course content profile. The conceptual framework involves learner profile, course content profile, the matching strategies between learner and course content profiles, and the initialization and update strategies of the profiles. The matching process is between the style of the learner and the type of the content. The style of the learner is configured using learning style theories. We model the learner according to three factors which are behavioral factors, knowledge factors, and personality factors. We modeled the course content using 30 content types. These content types are derived from previous studies, IEEE LOM Metadata[18] information, and the descriptions of Felder-Silverman[14] learning style model. We conducted a study to classify the content with the learning style and we defined the learning resources and the corresponding learning style dimensions as given in Table4.3. The adaptation strategy in the framework is to find the best match between the learner and the instruction set. Using the classification information, we find the best match between the learner profile and the course profile by applying a normalized Euclidian distance function.

In this thesis, we proposed a multi-agent adaptive system for online learning management systems, called MODA. The main components of the system are learner modeling, adapted content, and integration with LMS. We applied the conceptual framework developed in this study which defines the learner profile and the course content profile.

Since MODA was developed as a multi-agent system, we used Gaia as an agent-oriented software engineering modeling technique in order to model the system.MODA has seven agents: LMSInterfaceAgent, LearningAgent, ContentAdapterAgent, CourseProfileAgent, LearnerProfileAagent, ResearcherAgent, and AgentManager. LMSInterfaceAgent is the communication party with the LMS. LearningAgent is the central agent which is responsible for management of the other pedagogical agents. ContentAdapterAgent is responsible for finding the most appropriate content for the learner using the learner profile. CourseProfileAgent initializes and updates the learner profile. The agent deals with the course profile classification of the course content types. LearnerProfileAgent initializes and updates the learner profile. This agent updates the learner profile using learner actions. ResearcherAgent receives search results, communicates with ContentAdapterAgent and receives the adapted content. AgentManager is the creator of MODA environment. All of the MODA agents are collaborative pedagogical agents, and LMSInterfaceAgent is also an interface agent.

MODA agents work cooperatively to perform the operations such as login, search keywords, and view lecture notes. In the study, we provided the detail scenarios for these three operations.

The MODA were developed as JADE agents[19]. JADE (Java Agent Development Framework)[19] is a software development framework aimed at developing multi-agent systems and applications conforming to FIPA(The Foundation for Intelligent Physical Agents) standards for intelligent agents. Messages exchanged by JADE agents have a format specified by the agent communication language (ACL) language defined by FIPA (http://www.fipa.org) international standard for agent interoperability. In agent's communication, both the sender and the receiver must ascribe the same meaning to symbols for the communication to be effective. To achieve this, JADE defines ontology as the vocabulary of the symbols used in the content and their meaning[19]. In this study, we defined In six JADE ontologies in order to meet the requirements of the system. The ontologies are LearnerProfileOntology, LearningContentOntology, ActionHistoryOntology, LMSUserOntology, SeachCaseOntology, and AgentMessageOntology.

In literature, there are many adaptive systems providing powerful adaptive features. However, it is not possible to integrate most of these systems with the existing learning management systems. Most of them were developed to function as stand alone systems. There are lots of learning management systems used in practice and it might be very effective to plug adaptive features to these already existing and widely used learning management systems. In this research, the adaptive learning system was designed to be used with any learning management system. In order to achieve this, we defined a TCP-IP baed communication protocol between LMS and MODA. Any LMS providing necessary information with the required format becomes an adaptive learning management system, when it establishes a communication with MODA. Our protocol requires LMS and MODA to read/write the necessary information to TCP sockets. Since the communication occurs in the sockets, we defined the data formats exchanged between the systems during either requesting data or responding a request. In MODA, one of the agents, - LMSInterfaceAgent-, serves as a server that receives the requests from the LMS and provides the responses back. In the study, we provided the detail descriptions of the communication scenarios, request and response data types, and the structure of each data type used in the communication between LMS and MODA. MODA was implemented using Java programming language. Since the protocol requires communication through sockets, the development language of LMS becomes unimportant for the integration. This increases the usability of MODA in different LMS without considering the programming language limitations.

We have integrated MODA into a learning management system, called OLAT. Since OLAT is an open source LMS, we modified OLAT so that it can communicate with MODA. We created a sample course in OLAT and we provided different type of contents for the same course content item. We tested the system by logging as learners having different learner styles. The system gave the most suitable course content items regarding the type of learner. During this test, we had no problems with the communication protocol.

6.1 Limitations

There are several limitations while implementing and using MODA. These are as follows:

- Defining 30 content types for each content is a very demanding taks. To overcome this limitation, we suggest to extend the description of LOM metadata information so that it can provide to define 30 content types.
- The capabilities of MODA is higly related to the capabilities of LMS.
- Since we chose JADE as the middleware to develop agents, we had some problems while using the binaries of the middleware. At some time, the web site of JADE was down, and we had no access to the current binaries of the middleware. This caused wasting time.

6.2 Future Research

The effectiveness of the adaptive strategies and technologies are directly correlated with the number and variety of the learners. This means that the development of MODA is a never-ending process. There will be new adaptive features to be added, or new best matching strategies will be applied as we use it more. It provides a platform for further research.

In this study, we provided some suggestions for further practice:

- We applied a normalized Euclidian distance to find the best matche between the learner profile and the content profile. Different matching strategies may be applied and compared.

- We applied Felder-Silverman learning style model. It may be a good practice to apply different learning styles, and make a comparison on these applications.

- In our study, we used 30 content types to model the course content. One of the factors affecting the effectiveness of the adaptation is the content types. Therefore, it might improve the effectiveness and usability of the framework to conduct a study on understanding the validity and coverage of these content types in online learning systems.

- According to the architecture of MODA, the learner profile information is always kept in MODA. This provides us to use the learner profile data to study on and try to find some patterns for learner's learning in online learning environments. Thus, as a furhet study, oen can apply different AI techniques to learn the learner profile.

- MODA has been integrated to an open source learning management system, named OLAT[21]. As a further study, we are going to integrate MODA into the other learning management systems so that we establish a MODA community. The agents in this community might work collaboratively to analyze and learn the learner profile, and to provide personalized education.

- We have focused on asynchronous activities in LMS. It might be another future study to include learners collaborative learning tasks and synchronous activities into the process of adaptation

- We did not perform any validation or evaluation of MODA. The validation of the framework and the system can be conducted through the integrating MODA into a learning management system that have already been used with learners in an online course. We have already integrated MODA into OLAT. We have been in contact with OLAT community. We are planning to work collaboratively with OLAT community and to use MODA in the online courses given through OLAT. We have also provided the necessary issues to be considered in the evaluation study of MODA.

REFERENCES

[1] I. Park and M.J. Hannafin. Empirically based guidelines for the design of interactive multimedia. *Educational technology Research and Development*, 41(3):6385, 1993.

[2] P. Brusilovsky. Adaptive and intelligent technologies for web-based education. *KI - Kunstliche Intelligenz*, 13(4):19–25, 1999.

[3] P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: a tool for developing adaptive courseware. *Computer Networks and ISDN Systems*, 30(1):291–300, 1998.

[4] K. Nakabayashi, M. Maruyama, Y. Kato, H. Touhei, and Y. Fukuhara. Architecture of an intelligent tutoring system on the www. In *Proceedings of the 8th World Conference of the AIED Society*, Kobe, Japan, 18-22 August 1997.

[5] M. Specht. Ace - adaptive courseware environment. pages 380+. 2000.

[6] G. Weber and M. Specht. User modeling and adaptive navigation support in www-based tutoring systems. In C. Paris A. Jameson and Eds. C. Tasso, editors, *Proceedings of the 6 tn International Conference on User Modeling*, pages 289–300, Berlin Heidelberg, 1997. Springer-Verlag.

[7] J.M. Lpez, E. Milln, J.L Prez-de-la Cruz, and F. Triguero. Ilesa: a web-based intelligent learning environment for the simplex algorithm. In C. (ed.) In: Alvegrd, editor, *Proceedings of 4th International conference on Computer Aided Learning and Instruction in Science and Engineering(CALISCE-98)*, pages 399–406, Gteborg, Sweden, 1998.

[8] C. Wolf. iweaver: towards 'learning style'-based e-learning in computer science education. In *ACE '03: Proceedings of the fifth Australasian conference on Computing education*, pages 273–279, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[9] M. Grigoriadou, K. Papanikolaou, H. Kornilakis, and G. Magoulas. Inspire: an intelligent system for personalized instruction in a remote environment, 2001.

[10] M. Stern, J. Steinberg, H. Lee, J. Padhye, and J. Kurose. Manic: Multimedia asynchronous networked individualized courseware. *In Education Media and Hypermedia*, 1997.

[11] J. Gilbert and C. Han. Arthur: An adaptive instruction system based on learning styles. In *In Proceedings of International Conference on Mathematics / Science Education and Technology, AACE*, pages 100–105, 1999.

[12] R.A. Howard, C.A. Carver, and E. Lavelle. Enhancing student learning by incorporating learning styles into adaptive. In *In Proceedings of World Conference on Educational Multimedia and Hypermedia(ED-MEDIA)*, pages 118–123, 1996.

[13] E. Triantafillou, A. Pomportsis, and E. Georgiadou. Aes-cs: Adaptive educational system base on cognitive styles. In *Proceedings of AH2002 Workshop, Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, 2002.

[14] R. M. Felder and L. K. Silverman. Learning and teaching styles in engineering education. *Engineering Education*, 78(7):674–681, 1998.

[15] R. Mobbs. What is e-learning and how to become an e-tutor/e-teacher, 2007.

[16] W. Johnson, E. Shaw, A. Marshall, and C. Labore. Evolution of user interaction: The case of agent adele. *Intelligent User Interfaces*, pages 93–100, 2003.

[17] E. Andre, J. Muller, and T. Rist. The ppp persona: a multipurpose animated presentation agent. In *In Proceedings of the workshop on Advanced visual interfaces(AVI-96)*, pages 245–247, 1996.

[18] Ieee information technology, learning technology: Learning objects metadata lom. from the World Wide Web: http://ltsc.ieee.org.

[19] F. Bellifemine, A. Poggi, and G. Rimassa. Developing multi-agent systems with a FIPA-compliant agent framework. *Software: Practice and Experience*, 31(2), 2001.

[20] N.R. Jennings. Agent-oriented software engineering. In In Francisco J. Garijo and editors Magnus Boman, editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a*

Multi-Agent World:Multi-Agent System Engineering (MAAMAW-99), volume 1647 of *2*, pages 1–7, Heidelberg, Germany, 1999. Springer-Verlag.

[21] University of Zrich OLAT. Online learning and training, an open source learning management system, 2000.

[22] J.E. Kemp and G.R. Morrison. *Designing Effective Instruction (1 ed.)*. Macmillan College Publishing Company, New York, NY, USA, 1994.

[23] Daniel P.L. *Designing Effective Online Instruction: A Handbook for Web-Based Courses*. Rowman & Littlefield Education, 2006.

[24] C. Jung. *The Theoru of Psychological Type*.

[25] K. C. Briggs and I. B. Myers. *The Myers-Briggs Type Indicator*.

[26] Y. Gulbahar. Assessment of web-based courses: a discussion and analysis of learners' individual diffences and teaching-learning process. *PhD Thesis, Computer Education and Instructional Technology*, 2002.

[27] M. Catherine. The implications of the research literature on learning styles for the design of instructional material. *Australian Journal of Educational Technology*, 15(3):222–241, 1999.

[28] D.A. KoIb. *Experiential Learning:Experience as the Source of Learning and Development*. Prentice-Hall, Englewood Cliffs, N.J., 1984.

[29] R. Dunn. Understanding the dunn and dunn learning styles model and the need for individual diagnosis and prescription. *Reading, Writing and Learning Disabilities*, 6:223 247, 1990.

[30] K. C. Briggs and I. B. Myers. The myers-brigs foundation. web, June 2005.

[31] Campaign for Learning. What kind of learner are you?, 2007.

[32] R. Mobbs. What is e-learning and how to become an e-tutor/e-teacher, 2007.

[33] S.M. Parvez and G.D. Blank. A pedagogical framework to integrate learning style into intelligent tutoring systems. *J. Comput. Small Coll.*, 22(3):183–189, 2007.

[34] M. Zywno. A contribution to validation of score meaning for felder-soloman's index of learning styles. In *In Proceedings, 2003 ASEE Annual Conference*. American Society for Engineering Education, June 2003.

[35] R.M. Felder and B.A. Soloman. Index of learning styles questionnaire. accessed December 2006.

[36] L. Corno and E. Snow. *Adapting Teaching to Individual Differences among Learners In Wittrock, M. (Ed.) Handbook of Research on Teaching*. New York: MacMillan Publishers, 1986.

[37] R. Glaser. *Adaptive education: Individual, diversity and learning*. New York: Holt, 1977.

[38] J. Ong and S. Ramachandran. Intelligent tutoring systems: The what and the how. *Learning Circuits*, 2000.

[39] B. zdemir and F.N. Alpaslan. An intelligent tutoring system for student guidance in web-based courses. In *Lecture Notes In Computer Science. Proceedings of the First International Conference on Advances in Information Systems*, volume 1909, pages 437–448, London, UK, 2000. Springer-Verlag.

[40] L. Jerinic and V. Devedzic. A friendly intelligent tutoring environment - teacher's approach. *ACM SIGCHI Bulletin*, 32(1):83–94, January 2000.

[41] P. Brusilovsky. Adaptive hypermedia: From intelligent tutoring systems to web-based education. In *ITS '00: Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, pages 1–7, London, UK, 2000. Springer-Verlag.

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

[43] A. Paramythis and S.L. Reisinger. Adaptive learning environments and e-learning standards.

[44] D. Xu, H. Wang, and M. Wang. A conceptual model of personalized virtual learning environments. *Expert Systems with Applications*, 29(Issue 3):525–534, October 2005.

[45] C.S. Lin and M.S. Kuo. Adaptive networked learning environments using learning objects, learner profiles and inhabited virtual learning worlds. *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, pages 116–118, 2005.

[46] A. Davidovic, J. Warren, and E. Trichina. Learning benefits of structural example-based adaptive tutoring systems. *IEEE Transactions on Education*, 46(2):241251, 2003.

[47] M. Martinez and C.V. Bunderson. Foundations for personalized web learning environments. *ALN Magazine 4*, 2, 2000.

[48] M. Alavi. Distributed learning environments. *IEEE Computer*, 37(1):121122, 2004.

[49] J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering. *The tropos project, Information Systems*, 27(6):365389, 2002.

[50] A. Paramythis and S. Loidl-Reisinger. Adaptive learning environments and e-learinng standards. *Electronic Journal of e-Learning (ISSN 1479-4403).Academic Conferences Ltd.*, 2 (1):181–194, 2004.

[51] F. Mdritscher, V.M. Garca-Barrios, and C. Gtl. The past, the present and the future of adaptive e-learning.

[52] A. Bandura. Self-efficacy mechanism in human agency. *American Psychologist 37*, 122-48, 1982.

[53] *Agent Technologies, Infrastructures, Tools, and Applications for E-Services, NODe 2002 Agent-Related Workshops, Erfurt, Germany, October 7-10, 2002. Revised Papers*, volume 2592 of *Lecture Notes in Computer Science*. Springer, 2003.

[54] N.R. Jennings and M.J. Wooldridge, editors. *Agent technology: foundations, applications, and markets.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[55] J. Ferber. *Multi-Agent Systems An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, 1999.

[56] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[57] D.W. Croft. Intelligent software agents: Definitions and applications. web source, December 1997.

[58] TechTarget.com. web source, 1996-2000.

[59] N.R. Jennings, P. Faratin, T.J. Norman, P. O'Brien, M.E. Wiegand, C. Voudouris, J.L. Alty, T. Miah, and E.H. Mamdani. ADEPT: Managing business processes using intelligent agents. In *Proceedings of the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems (ISIP Track)*, pages 5–23, Cambridge, UK, 1996.

[60] R. Kozierok and P. Maes. A learning interface agent for scheduling meetings. In *IUI '93: Proceedings of the 1st international conference on Intelligent user interfaces*, pages 81–88, New York, NY, USA, 1993. ACM.

[61] H. Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[62] J. Davies, R. Weeks, and Revett M.J. Communicating information agents for www. Ipswich IP5 3RE UK, BT Laboratories, 1996.

[63] A. Chavz and P. MIT Media Lab Maes. Kashbah: An agent marketplace for buying and selling goods. In *Proc. of 1st International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, 1996.

[64] E. Shaw, W.L. Johnson, and R. Ganeshan. Pedagogical agents on the web. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pages 283–290, New York, NY, USA, 1999. ACM.

[65] A. Jafari. Conceptualizing intelligent agents for teaching and learning. *EDUCAUSE QUARTERLY*, (3):28–34, 2002.

[66] P. Suraweera. An animated pedagogical agent for sql-tutor, 1999.

[67] J.C. Lester and B.A. Stone. Increasing believability in animated pedagogical agents. In *AGENTS-97: Proceedings of the first international conference on Autonomous agents*, pages 16–21, New York, NY, USA, 1997. ACM.

[68] J.P. Bigus. The agent building and learning environment. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, pages 108–109, New York, NY, USA, 2000. ACM.

[69] Agentbuilder, 2000.

[70] The aglets mobile agent platform website.

[71] S. Poslad, P. Buckle, and R. Hadingham. The fipa-os agent platform: Open source for open standards. In *Practical Applications of Intelligent Agents and Multi-Agent Systems (PAAM 2000)*, Manchester, UK, The Practical Application Company, 2000.

[72] H. Jeon, C. Petrie, and M.R. Cutkosky. Jatlite: A java agent infrastructure with message routing. *IEEE Internet Computing*, 4(2):87–96, 2000.

[73] J.P. Bigus and J. Bigus. *Constructing Intelligent Agents Using Java.* John Wiley & Sons, Inc., New York, NY, USA, 2001.

[74] P.D. OBrien and R.C. Nicol. Fipa towards a stance for software agents. *British Telecom Technology Journal*, 16(3):51–59, July 1998.

[75] C.A. Carver, A.H. Richard, and W.D. Lane. Enhancing student learning through hypermedia courseware and incorporation of student learning styles. *IEEE Transactions on Education*, 1999.

[76] N.V. Stash, A.I. Cristea, and P.M. Bra. Authoring of learning styles in adaptive hypermedia: problems and solutions. In *WWW Alt. 04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 114–123, New York, NY, USA, 2004. ACM Press.

[77] T.Q. Sumner. Support personalization in distributed e-learning systems through learner modeling. *Information and Communication Technologies*, 1:610–615, April 2006.

[78] IMS. Learner information package specification v1. http://www.imsglobal.org/profiles, 2003.

[79] IEEE P1484.2/D7. Draft standard for learning technology. public and private information (papi) for learners. http://edutool.com/papi, 11 2000.

[80] N.R. Jennings. Agent-oriented software engineering. In In Francisco J. Garijo and editors Magnus Boman, editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, volume 1647 of *2*, pages 1–7, Heidelberg, Germany, 1999. Springer-Verlag.

CURRICULUM VITAE