Université de Montréal

Bimodal Adaptive Hypermedia and Interactive Multimedia a Web-based learning environment based on Kolb's theory of learning style

par

Bahram Salehian

Département d'informatique et
de recherche opérationelle

Faculté des arts et sciences

Mémoire présenté à la Faculté des etudes supérieures
en vue de l' obtention du grade de
Maîtrise ès sciences (M. Sc.)
en informatique

Décembre 2003

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé:

BIMODAL ADAPTIVE HYPERMEDIA AND INTERACTIVE MULTIMEDIA A WEB-BASED LEARNING ENVIRONMENT BASED ON KOLB'S THEORY OF LEARNING STYLE

Présenté par:

Bahram Salehian

a été évalué par un jury composé des personnes suivantes:

Petko Valtchev

President du jury

Esma Aïmeur

Directrice de recherche

Claude Frasson

Membre du jury

Mémoire accepté le 14 mai 2004

RÉSUMÉ

Ce travail porte sur le développement d'un système tutoriel intelligent pour le web, et d'un environnement virtuel dans un système hypermédia adaptatif. Il porte également sur l'application de la théorie de Kolb concernant les styles d'apprentissage. Le système se nomme BAHM (*Bimodial Adaptative Hypermedia and interactive Multimedia*), système dans lequel les utilisateurs peuvent s'exercer et apprendre par l'action (*learning by doing*). L'organisation du domaine utilisé dans BAHM supporte des cours individualisés, non-linéaires, cours qui sont reliés à un logiciel nommé SoftPhone, développé à Extendia Inc. Cependant, une fois que nous introduisons une structure non-linéaire, comme cela peut se produire dans tous les cours en ligne sur le web, le problème de « perte dans l'hyper-espace » peut survenir, ce qui introduit une confusion chez les utilisateurs : ils ne savent pas comment se souvenir où ils en étaient, et quel est le prochain sujet à étudier. Pour pallier à ces problèmes, des techniques de navigation adaptative sont utilisées pour aider et guider l'étudiant à travers la matière du cours. Originellement, la matière du domaine était présentée de façon à ce que chaque utilisateur reçoive le même contenu, sans prendre en compte les différences d'apprentissage de chaque apprenant. Cependant, BAHM peut considérer ces différences et adapter les informations à chaque utilisateur. L'adaptation du contenu est obtenue grâce à une approche comprenant deux phases, l'une considérant le niveau de compréhension de l'étudiant, et l'autre le contenu correspondant aux préférences de l'utilisateur. Un *Naive Bayes Classifier* est utilisé pour apprendre ces dernières en observant le type de contenu que l'utilisateur choisit de voir. Une étude empirique du système BAHM a été réalisée avec 8 utilisateurs, et a concerné l'installation, la configuration et la résolution des problèmes (« troubleshooting ») du logiciel SoftPhone. Les résultats de cette étude ont mis en évidence les différences dans les styles d'apprentissage des apprenants et ont permis de souligner le fait que l'utilisation d'une même stratégie d'enseignement pour tous les étudiants ne peut les aider efficacement : il est nécessaire de fournir une stratégie d'enseignement adaptée à chaque étudiant. Mais, dans certains cas, la technique de « learning-by-doing » peut avoir une incidence sur le temps passé à apprendre la matière aussi bien que sur la qualité de l'apprentissage.

Mots Clés

Systèmes tutoriels intelligents, Hypermédia adaptatif, Multimédia interactif, apprentissage par l'action, SoftPhone, Naïve Bayes Classifier, Environnement virtuel.

ABSTRACT

This work focuses on a web-based intelligent tutoring system and research issues associated with the development of a virtual environment in an adaptive hypermedia system and deployment of Kolb's theory of learning style. The system is BAHM (Bimodal Adaptive Hypermedia and interactive Multimedia), in which users can practice and learn-by-doing.

The domain organization used in BAHM supports a non-linear, individualized course, which is related to a software namely SoftPhone, developed at Excendia Inc. However, once we introduce a non-linear topic structure, as it might happen in all the web-based courses, the "lost in hyperspace" problem might arise, in which users become confused about how to remember where they have been and what to study next. To combat these problems, adaptive navigation techniques are used to help guide the student through the course material. The original domain material is presented so that each user sees the same content. This does not take into account learning differences of individual learners.

However, BAHM can consider those differences and adapt the information presented to each user. This adaptive content is achieved through a two phase approach which considers the user's level of understanding and the content that matches the user's preferences. A Naïve Bayes Classifier is used to learn the student's preferences by observing what type of content she chooses to see. An empirical study of the BAHM system was conducted with 8 users learning about installing, configuring, and troubleshooting SoftPhone. Results from this study show distinct differences in student's learning styles and provide evidence that using the same teaching strategies for each student cannot adequately support all students. But, in some cases, learning-by-doing can affect the time spent on learning the material as well as quality of learning.

Keywords

Intelligent Tutoring System (ITS), Adaptive Hypermedia, Interactive Multimedia, Learning-by-doing, SoftPhone, Naïve Bayes Classifier, Virtual Environment.

TABLE OF CONTENTS

CHAPTER I **1**

1. INTRODUCTION **1**

1.1 MOTIVATION 3

1.1.1 Benefits offered by networked systems *3*

1.1.2 How intelligent tutoring systems can affect the learning *4*

1.2 CONTRIBUTIONS 4

1.2.1 User modeling in adaptive hypermedia and multimedia systems *4*

1.2.1.1 How to make the adaptive presentation 5

1.2.1.2 Adaptive navigation and dynamic links 5

1.2.2 Wed-based educational system *6*

1.3 ORGANIZATION OF THESIS 7

CHAPTER II **8**

2. ITS LITERATURE **8**

2.1 DEFINITION OF THE ITS 8

2.2 GENERAL ARCHITECTURE AND COMPONENTS OF THE ITS 10

2.2.1 Student Model *11*

2.2.1.1 Learner Characteristics 11

2.2.2 Pedagogical Model *13*

2.2.3 Domain Model *13*

2.2.4 Communications Module *14*

2.2.5 Expert Model *14*

2.3 CONCEPTUAL TUTORS VS. PROCEDURAL TUTORS 14

CHAPTER III **16**

3. ADAPTIVE HYPERMEDIA AND MULTIMEDIA SYSTEMS **16**

3.1 ADAPTATION IN ITS AND ADAPTIVE HYPERMEDIA 17

3.1.1 Adaptive Navigation *18*

3.1.1.1 Link annotation 19

3.1.1.2 Adaptive link sorting 23

3.1.1.3 Direct guidance 25

3.1.1.4 Adaptive link hiding technique 28
3.1.1.5 Providing dynamic links 30
3.1.2 Adaptive presentation of the material *31*
3.1.2.1 Adaptive presentation based on conditional text 31
3.1.2.2 Alternatives 32
3.1.2.3 Stretchtext technique 35
3.1.2.4 Adapting everything else but the text 37
3.2 DISCUSSION 38

CHAPTER IV 40

4. TUTORING THE LEARNER 40

4.1 SPECIFICATIONS OF THE DISTANCE LEARNING 41
4.2 THEORY OF LEARNING STYLE 42
4.2.1 Hill's Cognitive Style Mapping *43*
4.2.2 Kolb's Theory of Learning Styles *43*
4.2.3 Dunn's Learning Styles *46*
4.2.4 Learning style preferences and adult Learning Styles *46*
4.2.5 What Motivates Adult learners? *47*
4.3 EXCURSION: THE BAHM VIRTUAL ENVIRONMENT 47
4.3.1 Availability of the Working Environment *48*
4.3.2 Electronic Communication Facilities *48*
4.3.3 Network Environment *49*
4.4 DISCUSSION 50

CHAPTER V 51

5. THE BAHM SYSTEM AND ITS ARCHITECTURE 51

5.1 OBJECTIVE 51
5.2 THE ARCHITECTURE 52
5.3 WHY WE CHOSE THIS ARCHITECTURE? 54
5.4 METHODOLOGY 56
5.4.1 Learning by doing *57*
5.5 MODELING THE KNOWLEDGE DOMAIN 59
5.5.1 Content objects *59*
Graphic objects: 59
Text objects: 60
Multimedia objects: 60

5.5.2 Topics *60*
5.5.3 Concepts *62*
5.5.4 Test questions *64*
5.6 MODELING THE STUDENT IN BAHM 65
5.6.1 What is represented in the student model *65*
5.6.1.1 Topics 65
5.6.1.2 Concepts 66
5.6.1.3 Student performances 67
5.6.1.4 Other sorts of knowledge in the student model 68
5.6.2 Using the student model *68*
5.6.2.1 Adaptive navigation 68
5.6.2.1.1 Rules for moving to a new activity 69
5.6.2.1.2 Computing "ready" scores 71
5.6.2.2 Adaptive content presentation 72
5.6.2.2.1 First phase – student's knowledge 73
5.6.2.2.2 Second phase – student's preferences 74
5.6.2.3 The tests 80
5.6.3 Updating the student model *82*
5.6.3.1 Timing information 82
5.6.3.2 Concepts 83
5.6.3.3 Content objects 84
5.6.3.4 Tests 85
5.6.3.5 The way regression equations were obtained 86
5.6.3.6 Student's performances 88

CHAPTER VI 89

6. GRADING & EVALUATION 89

6.1 GRADING TOPICS 89
6.2 TEST PERFORMANCE 89
6.3 STUDY PERFORMANCE 92
6.4 REVIEWED SCORE 94
6.5 COMBINATION OF THESE THREE SCORES 95
6.6 THE EVALUATION 96
6.6.1 Design *96*
6.6.1.1 The hypotheses 96
6.6.1.2 Experimental design 96
6.6.1.3 Results 97
6.6.1.4 Adaptation through user's instances 98

6.6.1.5 Timing and grading information 99

CHAPTER VII 102

7. CONCLUSIONS 102

APPENDIX A. REGRESSION EQUATIONS 106

APPENDIX B. BAHM APPLICATION DOMAIN - EXCENDIA 113

BIBLIOGRAPHY 124

LIST OF TABLES

TABLE 3-1 LIST OF SYSTEMS/TECHNIQUES 19
TABLE 5-1 FEATURES OF THE BAHM SYSTEM 59
TABLE 5-2 SAMPLE OF STUDENT RANKING 70
TABLE 5-3 LINK UPDATE RULES 72
TABLE 5-4 FEATURES IN BAHM SYSTEM 75
TABLE 5-5 A SAMPLE INSTANCE SPACE 78
TABLE 5-6 CONTENT OBJECTS IN CONCEPT *SWITCH* 78
TABLE 5-7 MAPPING BETWEEN MASTERED VALUE AND LEVEL OF DIFFICULTY 81
TABLE 6-1 UPDATING RULES FOR REVIEW 94
TABLE 6-2 LEARNED VALUES FRO THREE STUDENTS 95
TABLE 6-3. ACCURACY OF CLASSIFIER WHEN OBJECTS AT THE RIGHT LEVEL OF DIFFICULTY AND ASSUMING OBJECTS AT THE WRONG LEVEL OF DIFFICULTY WILL NOT BE WANTED 98
TABLE 6-4. USING DIFFERENT THRESHOLDS WHEN CLASSIFYING OBJECTS AT THE CORRECT LEVEL OF DIFFICULTY AND ASSUMING OTHERS TO BE NOT WANTED 99
TABLE 6-5 CORRELATIONS OF TIME SPENT STUDYING AND TEST PERFORMANCE 100

LIST OF FIGURES

FIGURE 2.1 INTELLIGENT TUTORING SYSTEMS LIE IN THE INTERSECTION OF THE COMPUTER SCIENCE, ... 9
FIGURE 2.2 INTERACTION OF COMPONENTS IN AN ITS 10
FIGURE 2.3 GENERAL FUNCTION OF AN ITS 11
FIGURE 3.1 SCHEMATIC VIEW ON ADAPTIVE HYPERMEDIA SYSTEMS 17
FIGURE 4.1 KOLB'S CONTINUUM RUNNING CYCLE 45
FIGURE 4.2 SNAPSHOT OF THE CONFERENCE ROOM IN BAHM SYSTEM 49
FIGURE 5.1 SYSTEM ARCHITECTURE 54
FIGURE 5.2 SYSTEM ARCHITECTURE IN THE SIMPLE WAY 55
FIGURE 5.3 SNAPSHOT OF THE BAHM'S VE (LEFT PANEL) 58
FIGURE 5.4 LINK-TYPE OF THE CONTENT OBJECTS 60
FIGURE 5.5 TOPIC STRUCTURE 61
FIGURE 5.6 TREE OF THE KNOWLEDGE NODES 62
FIGURE 5.7 A SAMPLE PAGE OF BAHM 63
FIGURE 5.8 TEST SAMPLE OF THE BAHM SYSTEM 64
FIGURE 5.9 EXAMPLE OF TWO REGRESSION EQUATIONS FOR THE SAME POINTS 87
FIGURE 6.1 HOW STUDIED VALUES ARE UPDATED BASED ON TIME READ 93
FIGURE 7.1 RETENTION RATES FOR DIFFERENT MODES OF LEARNING 104
FIGURE A 1 REGRESSION EQUATIONS FOR UPDATING THE LEVEL 0 MASTERY WHEN 106
FIGURE A 2. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 1 MASTERY WHEN THE STUDENT DID NOT STUDY OBJECTS FOR ENOUGH TIME 106
FIGURE A 3. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 0 MASTERY WHEN THE STUDENT STUDIED OBJECTS FOR AN AVERAGE AMOUNT OF TIME 107
FIGURE A 4. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 1 MASTERY WHEN THE STUDENT STUDIED OBJECTS FOR AN AVERAGE AMOUNT OF TIME 107
FIGURE A 5. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 0 MASTERY WHEN STUDENT STUDIED OBJECTS FOR TOO MUCH TIME 108
FIGURE A 6. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 1 MASTERY WHEN THE STUDENT STUDIED OBJECTS FOR TOO MUCH TIME 108
FIGURE A 7. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 0 MASTERY WHEN DONG THE ACTION WITH NO ERRORS 109
FIGURE A 8. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 1 MASTERY WHEN DONG THE ACTION WITH NO ERRORS 109

FIGURE A 9. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 0 MASTERY WHEN ANSWERING 4-ANSWER MULTIPLE CHOICE QUESTION CORRECT 110
FIGURE A 10. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 1 MASTERY WHEN ANSWERING 4-ANSWER MULTIPLE CHOICE QUESTION CORRECT 110
FIGURE A 11. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 0 MASTERY WHEN DOING THE ACTION WITH ERRORS 111
FIGURE A 12. REGRESSION EQUATIONS FOR UPDATING THE LEVEL 1 MASTERY WHEN DOING THE ACTION WITH ERRORS 111
FIGURE A 13. REGRESSION EQUATIONS FOR THE UPDATING THE LEVEL 0 MASTERY WHEN ANSWERING 4-ANSWER MULTIPLE CHOICE EQUATION INCORRECT 112
FIGURE A 14. REGRESSION EQUATIONS FOR THE UPDATING THE LEVEL 1 MASTERY WHEN ANSWERING 4-ANSWER MULTIPLE CHOICE EQUATION INCORRECT 112
FIGURE B 1 EXCENDIA AS AN ADJUNCT MEDIA SERVER TO EXISTING PBXS 116
FIGURE B 2. ACCESS & MANAGE THE FAX FROM OUTLOOK 118
FIGURE B 3. ACCESS & MANAGE THE VOICE FROM OUTLOOK 119
FIGURE B 4. VIEW THE INCOMING CALL 121

DEDICATION

…To my beloved parents

ACKNOWLEDGEMENTS

Although a dissertation can be a solitary activity, there are still many people who contribute to its completion. I would like to thank all those who have helped me complete mine.

I would like to start by acknowledging my thesis advisor, Professor Esma Aïmeur, for her supports and advise in directing my research efforts. She has been more than an advisor, but a friend to me. I would not have made it through graduation without your help, advice, and friendship.

I would also like to acknowledge the other members of my committee, Professors Claude Frasson, and Petko Valtchev for their help in completing this work. Thanks also to Professor Gilles Brassard for his enormous help all along my research. Also, I extend a very special thank to "Valorisation Recherche Québec" (VRQ), as this work was supported by them.

I would like to thank all those who have helped me at Excendia Company, those who gave me a framework within which I could complete this work. I would specially like to thank Mehrnaz Naghibi for all our conversations about my specific research and life in general. Thanks are also due to the eight users who participated in the study.

Lastly, I would like to thank my family: my brothers and my parents. You have all been with me since the beginning of this journey and supported me the whole way. I hope I've made you proud.

Chapter I

1.Introduction

Books, or, in former times, papyrus, leather, or slabs, have been the favorite holder of information since the invention of writing. The advantages of collecting information in books are various: information is grouped together, and often there is a red thread from the beginning to the end.

You can learn by reading a book or by discussing with a human – following the instruction of a tutor. The main difference in these scenarios is that a human partner or tutor will take the special learner into account and will adapt the learning speed and the depth of information to the vis-à-vis. A conventional book or learning instruction cannot adapt itself to the particular needs of its readers – written and printed once, it remains static. If we think of books and classrooms guided by the human tutor that are personally provided for us, we can suppose lots of useful ideas. For example, the books should not be boring by telling us things we already know or, we are not interested in. On the other hand, tutors should understand our abilities and our speed of learning and should be able to adapt them to our abilities. A tutor must constantly determine what to teach and how to teach. In order to make these determinations, the tutor must carefully monitor the progress of the student throughout the teaching session.

Think of systems, which demonstrate difficult and complicated topics by giving examples in our favorite and preferred ways. Think of systems which give a solid foundation of some topic and refer always – even after years – to the actual research in this area. Imagine an intelligent tutor which draws attention to itself if it has relevant information for you and provides explanations tailored to your actual knowledge state. Imagine an environment in which you can have the ability to experience what you want to learn without worrying about making mistakes that could harm you or your system.

Since the emergence of the World Wide Web (WWW) in 1991, the values of information and on-line tutoring have got a new dimension. Nowadays, millions of computers are connected via the Internet; humans can collect information from nearly anywhere in the world.

This enormous amount of information is also a chance for experience and learning. But effectively selecting information from the Internet is still a hot research topic as the effectiveness of search machines increase with the precision of the query. The information contained in the Internet is often useless for exploring or learning, as learners need guidance to build up a mental model of the area they are working on before being able to make sufficiently exact queries.

Not all the learners are computer scientists and expert in how to use the computer! It would be very helpful to have different learning methods centralized in a system for different types of learners, for students with different interests and different initial knowledge on the topic. To make a step in this direction, adaptive hypermedia systems personalize the content of the subject to the particular needs of users and an interactive multimedia environment embedded in those systems provides the ability of learning by doing and creating an experience [Brusilovsky, Stock, and Strapparava 2000]. They give users the ability to define their own learning goals, propose next reasonable learning steps to take, support project-based learning, give alternative views, and they can be extended by documents written by the learners. Adaptive hypermedia systems the repositories for accessing distributed information. Implemented as Internet applications, they can integrate and adapt these documents to the learner's goals and knowledge.

Promising approaches in research come from two areas, adaptive hypermedia systems, and interactive multimedia systems. Adaptive hypermedia systems combine hypermedia systems with intelligent tutoring systems to adapt the systems to the particular users. This work is a step towards intelligent environment based on an effective theory of learning style – Kolb's Theory. We propose concept and realization of an adaptation component combined with a virtual environment for an open, adaptive hypermedia system which, on the one hand, implements advanced teaching strategies, and, on the other hand, enables integration and adaptation of learning material to the learner.

This work is presented in the context of BAHM (Bimodal Adaptive Hypermedia and interactive Multimedia), a Web-based intelligent tutoring system. This system provides the students learn-by-doing inside a Virtual Environment (VE). It provides the

students with adaptive hypermedia content and an Interactive Multimedia as a virtual environment. In this system, students can read the course material, which is adapted to them individually. They may ask for help at any time, or even see the video help related to the specific topic.

1.1 Motivation

Web-based educational systems provide a good mechanism for distance learning, but the technology presents some difficulties. The Web makes interactivity difficult and provides a limited view of the student's activity. It is not possible to be aware of every action the student takes. Furthermore, we believe that learning-by-doing is one of the best and reliable methods of learning, but the problem is that, most of educational systems which support learning-by-doing are standalone systems which are not accessible through the Internet. The Web is a wonderful tool for hypermedia presentations. However, most hypermedia presentations are non-linear and users often find themselves lost within the hypermedia structure. Another problem is that, most of the material presented on the web is static; the same material is presented to all users. These limitations with the intelligent tutoring systems and hypermedia systems provide us with some of the motivations for the BAHM.

1.1.1 Benefits offered by networked systems

Distance learning is becoming more and more popular and prevalent and delivering learning systems over a computer network is more practical than distributing the software to all sites via CD-ROM. Specifically, each learning site or will not be required to have copies of the system for on-demand usage will not require users to wait for the system to be delivered. Network-based systems are available at any time and at any place around the globe. Because educational material can change rather frequently, using CD-ROMs for instruction is not cost-effective, since it requires redistributing the CD-ROMs every time the material changes. With a network-based system, instructional designers can

continuously upgrade and augment the material without any users being explicitly made aware of such changes. Therefore, networked educational systems allow for more flexibility and extensibility than traditional, static CD-ROMs.

1.1.2 How intelligent tutoring systems can affect the learning

It is becoming very evident that more efficient training systems are needed by all organizations undergoing rapid technological change. Lecture-style training and traditional instructional systems are unable to keep up with number of people who require training. New technologies are needed to reduce the increasing cost and burden of education and training. On the other hand, effective multimedia intelligent systems often include substantial multimedia components, making these systems memory and computer intensive. Because educational computational resources are typically limited, cross-platform delivery is essential. Thus, we again look to a Web-bases solution.

Properly designed computer-based tutoring systems have proven highly effective as learning aides. Intelligent tutoring systems (ITS) have been shown to teach twice as quickly as traditional classroom methods [Shute 1995] and produce increased skill retention with fewer mistakes.

1.2 Contributions

The contributions of the work presented in this thesis fall into three main categories:

- User modeling in adaptive hypermedia and multimedia systems
- Wed-based educational system
- Getting the benefit of interactive multimedia to learn-by-doing

1.2.1 User modeling in adaptive hypermedia and multimedia systems

This work contributes to both adaptive presentation and adaptive navigation

1.2.1.1 How to make the adaptive presentation

Most of the information found on the web is presented in a static and non-individualized manner; every user sees the same content. Similarly, most experiences students have with learning are also non-personalized. Teachers teach to entire classrooms of students at the same time, so all the students get the same material presented in the same way.

In order to overcome these problems, many software systems have been developed that adapt the content for each user based on characteristics of that user (a summary of these systems are available in [Brusilovsky 1998]). Many systems that adapt content have very simple user models. Some do not use dynamic user models at all. Our contribution in this work is not simply the use of a dynamic user model that adapts the content, but rather a user model that deduces information about students only by observing their behavior and performance. The system we have developed uses a two phase approach. We are interested in not only what the student should see, but also the way in which she prefers the material to be presented. We are concerned with students' modes of learning and we believe in learning-by-doing.

Simply presenting the "correct" information, but doing so in a suboptimal way for an individual student will result in fewer learning gains. Thus we ensure that the student learns the material by seeing it in the way she prefers to see it. We are also ensuring that the student is more satisfied with the course, since the material is presented in the way she likes it. This will lead to fewer frustrations for the student. Another contribution of this work is to include both time spent studying the material and the result of test to determine how well the student knows the material. We use a simple regression technique to incorporate both of these pieces of evidence. Since concepts have a level of difficulty, we keep scores on each of these levels. We then use these scores to determine the material that is at the most appropriate level of difficulty, to challenge the student but not to overwhelm her.

1.2.1.2 Adaptive navigation and dynamic links

Most adaptive Hypermedia systems support the user by providing guidance in the form of navigation tools. The disadvantage is that there is a higher risk for the students to get lost in this complex hyperspace, i.e. they do not know what parts the have already visited and

they are not sure what to visit next. Adaptive navigation helps students avoid this problem by providing guidance to students as to what paths to take through the course, based on reasoning about a student's learning need. Such techniques are typically based on suggesting paths through the material based on topics a user has already seen or how well she has been studying the last topic.

While we use a similar technique in BAHM, in that we base a suggested topic on what the user has already seen, our decision making process is more sophisticated. We do not equate viewing a topic with knowing a topic. Rather, we are interested in how the user is viewing material, specifically the amount of time spent, which version of material is seen, and if the student has to review the material. We are also interested in test grades as the most direct way to determine a user's knowledge. Most intelligent tutors and adaptive hypermedia systems use a single grade on a topic, both to determine whether or not the topic is known and if the topic should be studied. We use two different assessments for these values. We calculate how well a topic has been studied, which then enables us to determine whether or not the student is ready to learn other topics. The ready value is based on a student's performance on its pre-topics, which is the traditional topic score.

We also use a machine learning technique to get the adaptive hypermedia. Machine learning techniques are often developed and evaluated in domains that do not require fast, real time learning, since these techniques often require a great deal of training before the machine learner can accurately predict future events. However, if one desires to use machine learning techniques in user modeling, the machine learning technique must be fast and accurate. For this purpose, we have investigated the use of Naïve Bayes Classifiers for adapting the content the student sees. We demonstrate how a Naïve Bayes Classifier can be used to accurately predict uses' behavior. Although, the Classifier must learn the appropriate policy dynamically for each student it encounters.

1.2.2 Wed-based educational system

Since Web-based applications are potentially accessible from anywhere at any time, Web-based Learning Environments have been gaining increasing attention from the research community. The use of computer technology is essential to make education and training more efficient and effective. In this work we have merged a virtual training environment

with the appropriate adaptive navigation tools. In fact, we are representing two web-based modes of learning; learning-by-doing (learning by experiencing), and adaptive hypermedia.

We have used the web-based multimedia tools such as Macromedia Flash, to create the interactive environment in which student can gain practical experience without being worried about making any harm to the system. In the BAHM system we provided the student with the core presentational strategies, namely text-driven, audio-driven, and video-driven.

1.3 Organization of Thesis

In chapter II, I talk about ITS literature and its history. I introduce the adaptive hypermedia and multimedia systems in chapter III and I compare BAHM system with several other systems developed by other researchers. Chapter IV discusses related issues in different learning strategies and the one we have chosen to use. In chapter V, I present the BAHM's system, its features as well as its architecture and the reason we chose this architecture. In chapter VI, I introduce the modeling the knowledge domain, student model as well as the adaptation of the BAHM system. Grading and evaluation of the BAHM system is the topic of chapter VII.

Chapter II

2.ITS Literature

Despite the well-known and significant beneficial effect of individual tutoring compared to standard classroom instruction Bloom [Bloom 1984], one-to-one tutoring for all students is impossible. Ever since the computers are getting cheaper and more powerful people try to use them more than ever. One of the purposes of this kind is to getting higher education using personal computers. That is the most important reason of developing the computer-based instructional environment. The computer-based instructional environments which can adapt their teaching strategies to their users. In an attempt to bring (at least some of) the benefits of the one-to-one tutoring experience to a broader audience, researchers have produced computer-based intelligent tutoring systems whose goal is to mimic aspects of human tutors. We explore such systems with an emphasis on experimental evaluation and field trials. Researchers in cognitive science, computer science, and education have helped to establish the field of AI and Education.

2.1 Definition of the ITS

In the first glance we can say that Intelligent Tutoring Systems are computer-based instructional systems involving artificial intelligence with models of teaching strategies that specify how to teach, and instructional content that specify what to teach. In other words, Intelligent Tutoring Systems form a new and more advanced generation of Computer Aided Instruction systems. This field includes multidisciplinary research areas, such as artificial intelligence, education theory, psychology, cognitive science, and theory of human-computer interaction. One of their major features is their ability to provide a user-adapted presentation of the teaching material (see Figure 2.1). They suggest about the

student's mastery level of topics in order to dynamically adapt the content of course. Moreover, content models give Intelligent Tutoring Systems depth so that students can learn-by-doing in significant and practical contexts. Intelligent Tutoring Systems allow mixed-initiative tutorial interactions, where students can request questions and manage their learning. Using an expert pedagogue, instructional models make the computer tutor to more approach the benefits of personalized instruction. During the last decade Intelligent Tutoring Systems have came out of the laboratories and moved into classrooms and workplaces where mostly have shown to be highly effective, Shute and Regian [Shute, Regian, 1999]; Koedinger and Corbett [Koedinger, Corbett, Anderson, 1997]. When intelligent tutors are becoming more common and being increasingly effective, they are better to be accessible anywhere at anytime. Since web-based applications are potentially accessible from anywhere at anytime, the merging of the World Wide Web and ITSs augmented the effectiveness of such systems. Importance and benefits of web-based tutoring systems are comprehensible: platform independence, classroom independence, and easy deployment. As Brusilovsky [Brusilovsky, 1998] pointed out, web-based ITSs are intended to be used by a much wider variety of students than any standalone systems and many students may be working alone with web-based ITSs. So, to improve tutoring quality, web-based ITSs need to be improved more than ever.

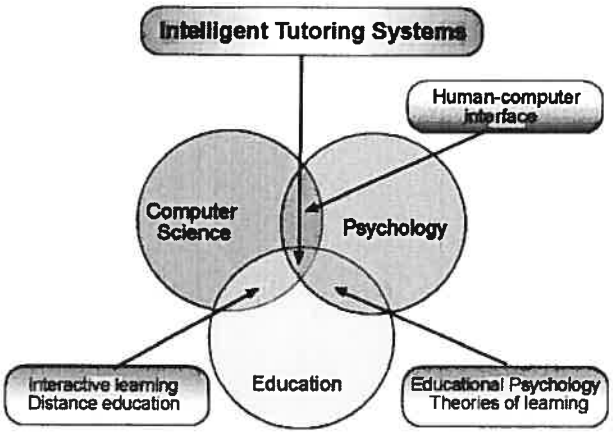


Figure 2.1[1] Intelligent Tutoring Systems lie in the intersection of the computer science, Psychology and education

1 taken from *Building intelligent* tutors Woolf, [Woolf, 2002].

We may enumerate the objectives of the Intelligent Tutoring Systems briefly as follows:

- To increase the students' participation in learning by making more effective use of the time they spend studying.
- To advance our understanding of how students learn difficult subjects by studying how different styles of tutoring change students' learning processes and outcomes.
- To advance the new methods for teaching fundamental concepts in the context of their application.

2.2 General Architecture and Components of the ITS

In a glance intelligent tutoring systems may appear to be gigantic systems, but from the design point of view, we can think about them as consisting of several mutually dependent components. Woolf has acknowledged four main components for the ITSs [Woolf, 1992]: the student model, the pedagogical model, the domain model, and the communication module. There is another idea that believes in a fifth component, the expert model, but Woolf considers this component as fraction of the domain model. Figure 2.2 provides a view of the interactions between the modules and Figure 2.3 shows how the basic structure of an ITS works.

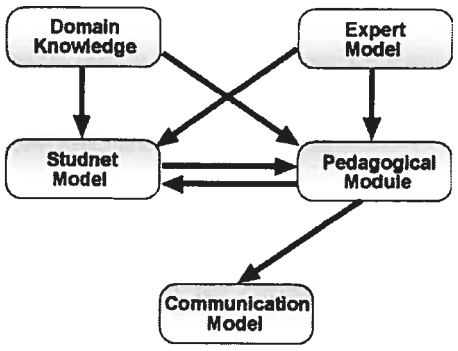


Figure 2.2 Interaction of components in an ITS

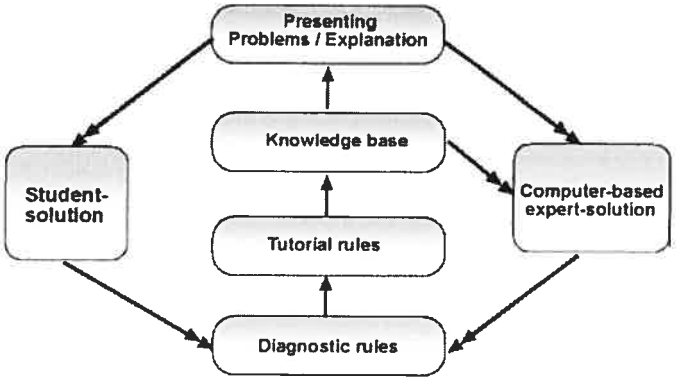


Figure 2.3[2] General function of an ITS

2.2.1 Student Model

Creating the student model engages defining the level of specialty in what the user background and history are, how the model should be obtained, and what the goals, knowledge and capabilities of the learner are. Information collected in the student model influences the tutor. The tutor can decide to advance through selected curriculum. It also helps the tutor to instruct or offer recommendation as well as to generate new problems or to assume set of explanations.

There are two common strategies to model the student's knowledge, *Overlay model*, and *Buggy model*. In the first case, the student's knowledge corresponds to a subset of the expert knowledge. Overlay model is generated by observation and comparison of the student's performance and system expert's performance on the same task. In the later case, intelligent tutoring system assumes everything as known unless the lack of knowledge is proven.

2.2.1.1 Learner Characteristics

As it is discussed before, adaptive hypermedia systems need information/data to evaluate the user. Peter Brusilovsky [Brusilovsky, 2001] identifies few features which are taken

into consideration by existing hypermedia systems. These features are enumerated as user's experience, user's knowledge, objectives, preferences, and background. We add another feature to this collection, as the time spent on problem solving.

User's Experience: This feature is about the hypermedia/multimedia experience of the user as well as her experiences about the subject being taught. If the user has worked with a hypermedia system before or if she has experienced working with a simulator, has she also worked with an adaptive hypermedia system?

User's Knowledge: User's knowledge is the major data for adaptation in the adaptive hypermedia systems or interactive multimedia systems. In particular, modifying of the knowledge state of a user is an important part for adaptation in instructive systems. The system keeps updating its assessment about the user's knowledge, and uses this information for selecting the next steps.

User's Objectives: Objectives of a user relies on her actual state of working with the system. Employing a hypermedia system and a virtual learning environment require different adaptation than a more overall learning objectives. There are two types of objectives for such systems, an overall learning objective as well as the problem solving task which might vary on each state of learning process.

User's Preferences: Using user's preferences, such as reading the text or watching the animation, pictures, or examples, let the system's users adjust their learning style. These characteristics could not be estimated by the system without receiving any input from the users.

User's Background: We consider all the user's experiences and knowledge which are not topic of the adaptive hypermedia system itself as her background. For example, experience and knowledge of a person who has worked as a network administrator in a company should be considered as a user characteristic and he cannot be considered the same as a person who is working in the marketing department of the same company.

---

[2] Taken from Ziemer Stephan

The time spent on problem solving: This user characteristic for educational hypermedia or multimedia systems and might be considered as another feature of a learner. Users with different learning speed should be treated differently.

2.2.2 Pedagogical Model[3]

This model offers the knowledge transportation in order to fit presentation of the teaching contents according to the student model. There are four major components which construct the pedagogical model. A module for selecting a concept to teach, a module which manage the order of the concepts to be taught, the module which is responsible for the teaching method, and a module for evaluation the user's performance.

2.2.3 Domain Model

Domain model consists of three parts: *knowledge concepts*, *concept groups*, *and course units* [Prentzas, Hatzilygeroudis, and Garofalakis, 2002]. This model consists of specifications, which determine the decision what material the system will present and when the system will present the material. There are two major methods of presenting the material:

A. Socratic Method: this method provides the students with questions guiding them through the process of debugging their own misconceptions.
B. Coaching method: it provides the student with an environment in which to engage in activities such as computer games in order to learn related skills and general problem-solving skills. The goal is to learn as a consequence of fun.

[3] taken from Jim Prentzas, Ioannis Hatzilygeroudis, and John Garofalakis 2002

2.2.4 Communications Module

Communications and interactions between the learner and the intelligent tutoring system are handled by this module. There is a matter of how the contents would be offered to the learner in the best efficient and useful way and should be set up to match the individual learner's level and proficiency.

2.2.5 Expert Model[4]

As we talked about earlier, the expert model and the domain model are almost similar and they both contain the information being taught to the learner. Though, the expert model is a model of an expert in a specific domain which represents the knowledge in a professional way. The development of the expert model may perhaps be the most challenging task. There are three common classifications of expert models, *Black Box*, *Glass Box*, and *Cognitive Modeling*. In the first case, the expert model is able to define whether a learner's answer is correct or not without explaining the reason. However, in case of Glass Box, a knowledge engineer collects the information from a skilled and expert in a particular domain and designs the computational representation of the knowledge. Finally, in Cognitive Model the expert model simulates not only the knowledge but also the way a human uses this knowledge. Although for a reasonable simulation of human problem solving processes, many research-questions must be answered.

2.3 Conceptual Tutors vs. Procedural Tutors

We may be able to classify the Intelligent Tutoring Systems into two major categories, Conceptual, and Procedural in which we use different learning methods. Depending on the nature of subject matter one of the two (or both) principle types of tutors should be chosen.

---

[4] taken from Ziemer Stephan

- *Conceptual tutors*, educate the student accurate knowledge and inferential skills. This kind of tutor is equivalent to a teacher in school.
- *Procedure tutors*, educate proficiency and procedures that have applications outside the tutorial situation. An example would be a computer-based flight simulator.

A combination of both tutors can be very functional as well. There are many differences between Procedural tutoring and Conceptual tutoring of which one is selection and sequencing of the knowledge representation. In the case of *Conceptual tutors*, the challenges are those of maintaining focus and coherence of covering the subject matter in an order that is the most practical for the student. The topic selection by conceptual tutors faces two restrains, subject matter, and tutoring the context. The conventional approach for conceptual tutoring is based on two major principles:

- Giving priority to the concepts that are more related to learner's knowledge
- Conversing the general concepts ahead of essential ones

The *Procedural tutors* have an additional dilemma to order the sub-skills of the specific skill and to select drills and instances to reflect that order. However, the main problem remains. In fact, none of the theories of learning is precise and powerful enough to support the choice of drills and examples in an ITS.

Chapter III

3. Adaptive Hypermedia and Multimedia Systems

The adaptive hypermedia and multimedia system expand the functionality of the hypermedia systems. The goal of our system is to individualize hypermedia systems to the individual users. Thus, each user has a personal view and individual navigational possibilities for working with the hypermedia system. Our system – BAHM – combines ideas from hypermedia systems, interactive multimedia systems, and ideas from intelligent tutoring systems. It fits in to the group of user adaptive systems, which are, for example, user adaptive interfaces or user model-based interfaces [Jameson, 1999]. BAHM system uses a user model to collect information about her knowledge, goals, experience to adapt the content and the navigational structure. Let us have a look at an example. For a user with little knowledge it might be useful to read more introductory information before going into detail. However, the same information would not be so interesting for a more knowledgeable person. Here, the choice of the right information at the right time is the task of the user model. The support of the adaptive system is useful if there is a common system which provides many users with different goals, knowledge, and experience, and if the underlying hyperspace is relatively large [Brusilovsky, 2001]. The user's aims and knowledge can be used for limiting the number of available links in a hypermedia system. Regarding the definition of adaptive hypermedia systems, Peter Brusilovsky proposes [Brusilovsky, 1996]:

> "*By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user.*"

The ideal applications of adaptive hypermedia or multimedia systems are educational hypermedia systems where the user or student has a certain learning goal. In BAHM system, the center of attention is on the knowledge of the students, which may differ extremely and the knowledge state changes during the work with the system. Thus, a correct user modeling of updating knowledge, and the ability to make the right decision on base of the updated knowledge estimations are the significant parts in BAHM system.

3.1 Adaptation in ITS and Adaptive hypermedia

An adaptive hypermedia system brings together information about users. It adjusts its content and navigational possibilities to the particular user, based on those individual characteristics (figure 3.1).

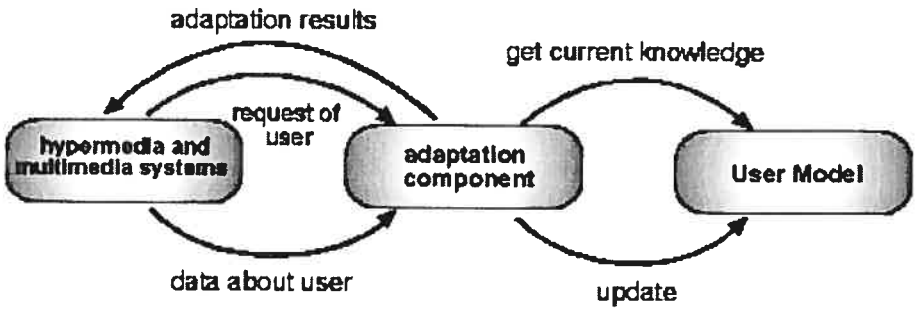


Figure 3.1 Schematic view on adaptive hypermedia systems

Hypermedia systems are systems in which there is a non-liner progression through the material. Students have many branch points and can choose to follow any of many possible paths. Adaptive hypermedia systems add an intelligent component to hypermedia systems to help users find their way among those paths. There are two general techniques used for adaptive hypermedia systems: adaptive navigation and adaptive presentation [Brusilovsky, 1998] [De Bra, 1998]. Adaptive navigation is a method to alter the way a student progress through a course while adaptive presentation is a process to change the actual material the user sees.

3.1.1 Adaptive Navigation

Adaptive navigation skills are intended to help a user proceed through an existing hyperspace. It does not alter the content of what the user observes; fairly it maintains the user in her navigation through the gap. There are plentiful methods for carrying out adaptive navigation including link annotation, link sorting, direct guidance, link hiding, dynamic link, and adding new links. In BAHM, we use both direct guidance (through the "next" button) and link annotation (through a tree-view table) as well as dynamic links. By using link adaptation, the user's possibilities to navigate through the hypermedia system are personalized. The following methods show examples for adaptive navigation support.

Link annotation: Interpret the links to give the user hints to the content of the pages they point to. The interpretation might be text, coloring, an icon, or dimming.

Adaptive link sorting: Sort the links of a document suitable to their significance for the user. Comparison sorting, precondition knowledge the "significance" of a link to the user is based on the systems hypothesis about the user. Several systems sort links depending on their resemblance to the present page, or by ordering them according to the required precondition knowledge.

Direct guidance: Guide the user sequentially through the hypermedia system. The following two methods can be distinguished:

- *Next best:* Provide a "next" button to navigate through the hypertext.
- *Page sequencing:* Spawn a reading succession through the entire hypermedia system or through some part of it.

Adaptive link hiding method: Limit the navigational possibilities by hiding links to inappropriate information. Hiding of links can be realized by making them unavailable or invisible.

Providing dynamic Link: Redirect the user to the specific page based on the information obtained from user. In this case we do not hind the link, but we make the link adaptive to the user's knowledge and preferences.

Link adaptation limits the quantity of links and thus the number of navigational possibilities. It is useful to prevent the user from getting lost in hyperspace. As in the case of content level adaptation, a description of the content of the documents is required for implementing the adaptation tasks.

Table 3-1 List of systems/techniques

	Link annotation	Adaptive link sorting	Direct guidance	Adaptive link hiding	Dynamic links
ELM-ART	Y				
ELM-ART II	Y				Y
InterBook	Y				Y
KBS Hyperbook	Y		Y		Y
ACE	Y		Y	Y	
WebWatcher	Y				
AHA	Y			Y	
MetaLinks	Y				
Anatagonomy	Y				
RBPR	Y				
HyperMan		Y			
TANGOW			Y		
ADAPTS	Y		Y		
C course			Y		
HYNECOSUM				Y	
DynaWeb					Y

3.1.1.1 Link annotation

One of the most common forms adaptive navigation support is to annotate the links that are presented to the user. These annotations provide the user with a guide for which nodes in the hyperspace to visit next.

ELM-ART [Brusilovsky, Schwarz, and Weber, 1996] was one of the first systems to use link annotations. This system was an intelligent tutoring system for teaching LISP,

and consisted of a textbook and a reference manual. ELM-ART used both an icon (red, green, or yellow ball) and different font (italic, bold, or plain) to designate the state of the content of the link. Red meant the content was not ready to be learned, yellow meant the page was ready to be learned, and green meant the content was known.

ELM-ART II [Weber, Specht, 1997] continued with the same metaphor as ELM-ART, but completed its functionality. This system used a green ball to indicate that a page was ready to be visited, a red ball to indicate that the page was not ready to be visited, a yellow ball to point out that a test had been solved, the page had been visited, and an orange ball to show either that the system had indirect the content of the page to be known or that the page had been visited but not all the sub-pages had been visited or worked effectively. This system also used direct guidance by calculating the best next step the student should take. Ongoing in the line of ALM-ART and ELM-ART II, InterBook [Brusilovsky, Schwarz, 1997] was a common atmosphere that supported the creation of intelligent, adaptive books. This system also used both direct guidance and adaptive annotations to guide students through the course. Annotations were talented through four colored balls and three fonts. The annotations were time-based, information-based, and precondition-based. The user model was started through a registration page via stereotype models. A study of the InterBook system [Brusilovsky, Eklund, 1998] showed that most students did not take advantage of the link suggestions. Most students used the "next" and "back" buttons, which did not have annotations. In fact, more than eighty percent of the navigation choices were made without annotations. Furthermore, there were no statistical meaning differences in test scores between the group with adaptive navigation support and group without it. However, those students who had adaptive navigation support used more non- ordered features. One application of the InterBook approach can be found in the ACT-R bookshelf [Brusilovsky, Anderson, 1998]. A bookshelf is a collection of several books on the same subject. The ACT-R student model used both a student's visit to a page on a concept and the student's answer about a concept on a test as evidence that the concept was known. The second kind of evidence, test taking, was considerably stronger and in ACT-R, a concept could not be well-learned until such evidence was provided.

The KBS Hyperbook [Henze, Nejdl, 1999] system also used the traffic light metaphor for link annotations. The annotations were placed on links between information units, which are the semantic parts of the course. The annotations indicated that an information item was already known, suggested, or too difficult.

ACE [Specht, Opperman, 1998] was an adaptive courseware setting on the WWW. The system used an amount of adaptive navigation methods: adaptive annotation, incremental linking, and adaptive sequencing. Links were annotated with colored balls based on learner's knowledge state and a pedagogical model of the domain units. A red ball indicated a concept the student was not ready to learn, a green ball indicated a concept ready to be learned, and an orange ball indicated units that were not missing any prerequisites but were still not recommended. Furthermore, links pointing to concepts the student had visited had hook icon annotations.

Langenbach and Bodendorf [Langenbach, Bodendorf, 1997] present a system that also used annotations to help guide users. The interface for this system consisted of two frames, one for content and one for navigation. The navigation buttons included Next, Up, and Previous. The next button took a student through predefined guided tour; the up button took her to the root of the hierarchically preceding module; and the previous button went back one step in predefined tour, even if the user had lost the path. A user could also use a table of contents, overview pages, or follow links for an unstructured tour. Color coding and dynamically updated tables of contents were used for orientation guides.

WebWatcher [Armstrong, Frietag, Joachims, and Mitchell, 1995] also provided link annotations to users to suggest which ones they should follow. WebWatcher was an interface added on top of existing websites. The user started her interaction with WebWatcher by providing her information looking for goal. The original page the user was viewing was then altered, after being processed by WebWatcher. WebWatcher made suggestions for links to follow based on the probability the user will want to see the link, which is a function of the current page, the goal, and the link. Machine learning techniques were used to determine which best links to suggest to users.

Kaplan et al. [Kaplan, Fenwick, and Chen, 1998] discussed a way to rank topics that were relevant to the current topic and to the current goal. An associative matrix was used to store the degree of relationship between two nodes. Users could indicate their preferences for good relationships, and this changed the associative matrix. It was also possible to combine matrices for multiple users to create rules for classes of users. The system could also learn association strengths by viewing how long users read a topic, which was normalized by how much information is presented in the topic. The longer the viewing, the more relevant the topic was. The authors acknowledge that it is still hard to determine if the information was read or not.

AHA [De Bra, Calvi, 1998] also used link annotations to indicate pages that are desired (blue), uninteresting and /or visited (purple), and undesired (dark gray). For each page, the author must determine a Boolean expression of variables that described whether the page was desirable or not. The system could compare the user's model to this expression to determine if a page was desirable to that user, and annotate it accordingly. The desirability of a link was dependent solely on the destination of the link, not on the source. The system presented in [Geldof, 1998] used natural language techniques for link annotations. The topics in the system were modeled hierarchically, and the goal was to determine which topics were of interest to the user. The topics concern features of movies, such as the time and place it is being shown, the actors, the director, or the plot. The system used as evidence of the user's interests her clicks in the topic pane and her concrete queries. The system used template based natural language generation to generate meta-text, which provided details on the topics in which the user had shown an interest, instantiated for this movie.

MetaLinks [Murray, Shen, Piemonte, Condit, and Thibedeau, 2000] used link annotation, not to suggest which next links to study, but to provide an orientation to the user. The table of contents showed students which nodes they had visited and indicated where the student was within the content hierarchy. Students could also see a history of where they have been annotated with the type of link.

SiteIF [Stefani, Strapparava, 1998] attempted to automatically rank the relevance of web pages based not on the user's direct feedback, but on which pages were requested. A user profile was built and updated after each request. The user model was a semantic net with the nodes being words and the arcs the co-incidence relation of two words. This model was used to decide if a document was valuable or not.

Annotations can also be used with news retrieval systems. The Anatagonomy newspaper on the WWW [Sakagami, Kamba, 1996] attempted to infer a user's preferences and use these preferences to retrieve relevant news articles. The score engine was used to compute the importance of each article by comparing the article to the user model, which was composed of a set of words and their weights. The learning engine built the user profile based on explicit feedback given by the user. Each article was presented with a score bar indicating the score computed by the score engine. The user could adjust this score to reflect her true interests. When this happened, the learning engine updated the user profile based on this new score. Letizia [Lieberman, Letizia, 1995] was an agent that

supports the web browsing by locating and annotating links in which the user may be involved. Letizia modeled the user's browsing behavior and used this to infer her interests based on the links followed and the pages to which the user has returned. Links that have been unobserved were assumed to be less interesting. Using this information, Letizia searched from the user's current browsing point to find web pages to recommend.

Most of the link annotation systems for data repossession used explicit modeling, i.e. the user provided feedback on whether pages were good or bad. Kushmerick, et al. [Kushmerick, McKee, and Toolan, 2000] presented a technique for Referrer Based Page Recommendation (RBPR), and implicit technique. This technique made page recommendations based on browsing behavior only. It assumed that a visitor's true information need could be characterized by list of query terms, and RBPR guesses these terms by looking at the page that led the user to the site. The authors claim their system is seventy seven times more effective than random guessing. Goecks and Shavlik [Goecks, Shavlik, 2000] also presented a technique to automatically learn user's preferences and interests through purely explicit means. With this technique, the system foresaw the percentage of hyperlinks clicked on, the amount of scrolling a user did, and a user's mouse activity. From these systems, we can see that link annotations have many uses and many instantiations, such as simple graphics to more detailed natural language comments. These annotations can provide users with help for choosing which links to follow. Furthermore, since these recommendations may not be correct, the user has the option to ignore them, since the links are still enabled even if they are not recommended.

However, annotations can also be confusing, especially if the user does not know how the annotation was computed. Similarly, an annotation may be overloaded with different meanings under different circumstances. If annotations are to be used, they should be simple in their presentation so users can quickly view them and understand the recommendations.

3.1.1.2 Adaptive link sorting

Link sorting is a technique that presents links to users organized from most recommended to least recommended, as computed by the system, based on the user's behavior or knowledge. The system presented in [Bollen, Heylighen, 1997] adapted the weights on nodes based on many users' prototypes. The nodes with the highest weights were then

presented to the user in sorted order. The authors point out the flaw that those nodes with the highest weights will continue to have the highest weights, since those are the ones presented to users as the top nodes. Adaptive HyperMan [Rabinowitz, Mathe, and Chen, 1995] sorted links based on significance. In this system, users could mark parts of pages and assign topics to those markers. This repossession was done either through exact comparison or through derivation of previous queries which contained subsets of the current query. The markers that were returned were sorted by a significance measure obtained through the significance network, which was constructed through user feedback. Users provided either positive or negative feedback for marker retrieval and this feedback was used to update the relevance network.

Ardissono and Goy [Ardissona, Console, Torre, 1999] presented work on sorting news articles on an individual basis. In this system, stereotype user models were used, which were derived from an initial form filled put by the user. The system used this model to select appropriate sections to determine the detail level for presentation of the news and advertisements, and to dynamically generate the page. The user model was dynamically updated based on certain events, such skipping some sections or news, following a link looking for more detail, restraining some of the detail, and selecting a banner ad. The system also took into consideration the number of connections to each section per week and/or month.

Just as with link annotation, link sorting leaves all link enabled, which allows users to rule against the judgments of the system. Furthermore, users can quickly observe the systems recommendations since the most highly suggested link are at the top of the link. With link annotation, this may not be the case and the user would have to search the entire link and read all the annotations to find the best links. However, with link sorting, we sometimes see a problem that arises with link recommendations based on popularity. In this case, the most popular links based on visits are presented at the top of the sorted list. These links then continue to be the most followed and thus the most popular, even if they are not the best links for the user's situation. A better link may be buried in the middle of the list and this link may not become popular because of that placement. Thus users may never be directed towards better links.

3.1.1.3 Direct guidance

Direct guidance is a technique that provides the user with a view of the hyperspace so that she can better decide where to go. With direct guidance, a sequence of links or a path is usually generated for the user to follow through the hyperspace. This is different from link annotations or link sorting since with those techniques, links are treated and judged individually. Direct guidance considers the set of links presented to the user as a whole unit, not as a list of individual links.

The KBS Hyperbook [Henze, Nejdl, 1999] used direct guidance to help a user effectively complete a course. In this system, sequential trail was to be generated for each user, ordering the content in a rising level of difficulty. In this system, the student model was represented as a vector over the knowledge items, where a knowledge item was a knowledge concept of the application domain. A Bayesian network was used for calculating the system's belief of the user's knowledge on each knowledge item. There were four possible grades on knowledge items, expert, advanced, beginner, and new comer, determined by either direct feedback from the user after working on a project – exercises, or examples of solved problems – or the judgment of an expert. No information was collected through the user's reading behavior or the system's assessment of the student's performance on a project. When generating a trail, the system performed a depth-first traversal to check the knowledge items that were prerequisites for the user's current goal. It then generated a list of those knowledge items not yet satisfied, ordered increasing difficulty.

Rivlin et al. [Rivlin, Botafogo, and Schneidermann, 1994] presented a way of viewing already existing hyperspaces so that they were easy to navigate by users. Questions such as "where am I" and "how do I get somewhere" could be answered if the current node could be put into some kind of context with respect to the rest of the hyperspace. One technique tried to determine the root of the hierarchy. Clusters were formed for ease of understanding. Nodes were rated based on how many links go in and how many go out. Rather than provide a simple list of domain topics, [Zyryanov, 1996] presented a method using an adaptive map of the domain. How much of the map was seen was based on the student model. The nodes that were presented were annotated with suggestions by the tutor. Case-based reasoning has also been used for guidance in adaptive navigation.

Micarelli and Sciarrone presented such a system in [Micarelli, Sciarrone, 1996]. This paper proposed a technique for hypermedia navigation based on case-based reasoning. A neural net was used to classify which canonical path a user was using in order to meet a learning goal. The canonical path from the case library was determined based on the initial nodes visited by the user. When help was requested, the system suggested the next steps for navigation, by building a hierarchical tree whose root is the last node of the canonical path.

In TANGOW [Carro, Pulido, and Rodriguez, 1999], tasks were decomposed into sub-tasks in multiple ways, depending on the student profile. Direct guidance for achieving tasks was provided through a Task Manager. This Task Manager stored data about the student's actions and results in a dynamic tree. The nodes in this tree corresponded to the tasks achieved by the students, and the edges represented how the tasks had been decomposed during the learning process. At each step in the learning process, the Task Manager constructed a list of attainable tasks, which were tasks that were in the set of sub-tasks in which the current task is decomposed, had not yet been achieved, and had an activation condition that was satisfied. Tasks which did not meet these conditions were listed, but links to them were not provided. When a task was completed, the Task Manager would search up through the dynamic tree to find a new task to compete, and used the same algorithm to construct a list of achievable tasks.

The ADAPTS system [Brusilovsky, Cooper, 1999], as one of its many adaptive techniques, used direct guidance to help technicians accomplish their tasks. The system presented step-by-step directions, customized to the user's troubleshooting experience. The system presented an adaptive task checklist which suggested an optimal path. This checklist also indicated the current state of performing the task. Annotations were also used to indicate completed, current, and remaining steps in the task.

The ACE system [Specht, Opperman, 1998] also performed adaptive sequencing of the course content using different sequencing algorithms for different teaching goals. ACE adapted a sequence of whole learning units and adapted the sequence of media available for a unit. The goal was to keep a student on her optimal path based on her current knowledge and to adapt the sequence to the interests specified by user. One novel approach with this system was that it computed an appropriate teaching strategy and generated a personalized view on a learning unit, based on learner interests and learning material preferences. ACE also automatically adapted to the learning strategy frequently

requested by student, especially if style led to success. When a certain threshold was reached indicating that this style was preferred and was successful, it was used as default. In order to achieve this adaptation, the system kept track of various information of the student, including her preferences for language, media, interface setting, and personal annotations, all of which were updated through a direct user dialog. The system also recorded confidence values on learned units – the tested, requested, and inferred confidence. The student model also contained inferences about the learner's interests.

A more direct approach to guiding a student through a course can be taken using direct guidance. By altering the contents of the "next" link in a hypermedia system, the system itself can direct guide the student through the material. This guidance is typically based on the user's understanding of prerequisite topics. We use this technique in BAHM for dynamically deciding the results of using the "next" button.

The C course in [Kay, Kummerfeld, 1994] used direct guidance when constructing the "next" links which appeared on each page. The system used if-then rules to determine which link, among possible alternates, should be included in the student's page. This was essentially conditional text applied to adaptive navigation. The *if* part of the rule was compared to the user model, and if it evaluated to *true* the *then* part of the rule fired. Direct guidance is a technique that is similar to link sorting in that suggested links are presented to the user in a preferred order. However, link sorting only presents the links individually and not as a complete path, and thus after the user follows the first link, he may not know the rest of the path. Therefore, direct guidance that can be more beneficial than link sorting since users are presented with more comprehensive suggestions in terms of a path to follow, rather than just the next link. However, direct guidance can be confusing for users since they may distinguish that the structure of the hyperspace is changing beneath them. For example, a user viewing topic A asks for guidance. The first time the user views the topic, the system may guide the user to view topics B, then C, and then D. But the second time the user views topic A, the system may guide the user away from topics B, C, and D since he has already seen them, and towards topic E. Thus the user will not see the same structure to the hyperspace under these different circumstances. She does not see consistent links between topics in the hyperspace.

3.1.1.4 Adaptive link hiding technique

Adaptive Link hiding is a technique in which links to part of the domain are hidden form the user until the tutor decides she is ready for the contents. The links that are hidden exist in the hyperspace structure, and are simply inaccessible until the system makes them accessible. A popular approach for hiding is not to allow users to see links unless all the prerequisites have been viewed. One example of this is in [Calvi, De Bra, 1997], which described a system in which users could only access certain links if they have seen all the appropriate precondition contents. A domain expert entered a link structure, including all possible interesting links. Dynamically, which links were interesting were determined based on a user model. Additionally, links could be taken away; if it was determined the user no longer needed them. A second example can be seen in [Pérez, Gutiérrez, Lopistéguy, 1995]. In this system, users were categorized as novice, medium, and expert, based on their year in school. Exercises were used to help the system judge the student's abilities. Curriculum Decision Rules were used to decide which new concepts to let the student see based on pedagogical relationships, difficulty level, known concepts, and student learning characteristics. Another system that used link hiding is AHA [Pilar de Silva, Van Durm, Duval, and Olivie, 1998]. The main components of this system were a domain model, a user model, and an adaptive engine. The links in hyperspace were weighted and the nodes were also typed. Nodes could be either documents or concepts, and there could be links between concepts or between concepts and documents. Each document in the space had a level of difficulty. When a user viewed a document, the associated concept's knowledge level was updated, if the document's level of difficulty was greater than the concept's level. Links to relevant documents were accessible to the user, if they had difficulty level considered to be appropriate for the user. Other links were not visible.

AHA [De Bra, Calvi, 1998] also used link hiding as its adaptive navigation techniques. In this system, when a link was hidden, the anchor text was still available, and the link still active, so if user knew this, she could click on the link. The system also supported link removal through conditional content.

HYNECOSUM [Vassileva, 1998] took a task-centered approach to link hiding. In this system, users were classified via stereotypes based on a user class, such as doctor, nurse, administrator, student, or patient. These stereotypes initialized the experience levels

of users on tasks in the domain. Tasks were defined hierarchically, and the individual user models were overlays of this task hierarchy. Users explicitly selected tasks to perform, so the system did not have to suppose goals. The user's access to the task hierarchy was restricted by her experience level on the task. Users could not access tasks for which she was a novice; she must first select sub-tasks before viewing the task itself. Furthermore, beginner users were limited to restricted browsing, in which she could only view entities directly related to the current task. More advanced users had the right to perform free browsing with an anchor, i.e. they could explore more of the hyperspace. Links between hypermedia entities had values ranging from –2 to 2, which were computed from the user's model. If, during navigation, patterns emerged that the user's skill on a task had increased, the system would ask the user if she wanted to increase her skill rating in the user model. Thus the system was adaptive, but only with the explicit consent of the user. These patterns involved the user's navigation through the task hierarchy; a minimum time spent on an information item, and correct attempts to select from the semantic classification menus. Another approach to hiding is to hide documents, rather than links.

The system in [Gonschorek, Herzog, 1995] used this technique. This paper described an adaptive help system that was part of an intelligent tutoring system on parallel programming. Users were categorized as novice, beginner, advanced, qualified, and expert. Each document in the help system was related to document classes. There were three document classes, the first of which described the knowledge the student should have, the second identified the lecture, and the third described the kind of text the documents were (information, introduction, outlook, etc.). The technique used in this system hid entire document classes, rather than single documents. Links to hidden documents were simply not presented to the student. In addition to the techniques previously described in the ACE system [Specht, Opperman, 1998], the system also used link hiding, which the authors called incremental linking. In this technique, added elements of the interface and tasks were incrementally introduced to a learner based on difficulty, complexity, and already mastered objects. When the user worked with the system, it would present links to already mastered concepts and ready-to-learn concepts. Furthermore, the system disabled links for not ready to be learned units. From these systems that use link hiding, we can see that the technique is used to prevent users from seeing material for which they are clearly not ready. If the system can accurately judge this, then this technique may be beneficial for users since they will not become irritated

reading material beyond their level of understanding. However, if the system overestimates a user's knowledge, and she is ready for that material, there is no way for her to reach it since the links have been removed. Therefore, unlike many of the previous techniques, users cannot recover from a system's mistakes.

3.1.1.5 Providing dynamic links

The technique of providing dynamic links that do not necessarily exist in the hyperspace allows a dynamic structure to emerge, based on individual differences. Thus the structure of the hyperspace will be different for each user. Kushniruk and Wang used this technique in [Kushniruk, Wang, 1994]. This system adaptively added links to a node that would not normally have been connected to this node. New links were added based on the student's performance on exercises. Links were added if they addressed misapprehension held by the student. If a link was added to a node, it was immediately presented as an option to the student to view that newly linked node. Thus, paths through the hypermedia space were different for each student. The DynaWeb system also added dynamic links on the fly [Bodner, Chignell, and Tam, 1997]. In this system, there were no static links. Rather, dynamic links were created based on the content of the website and the interests of the user. The selection of words used for links was based on previous links user has selected. The system also used a running query which was constructed using the surrounding text and text from the two previously followed links. This information was also used to build a user profile. The result from a study with DynaWeb show that dynamic linking improved question answering performance, mainly for novice users not familiar with search domain. Novice users also had a reduced task time.

The KBS HyperBook project [Henze, Nejdl, 2000] dynamically computed links between related information. The system was designed to be "open" in that it could include outside material. Courses consisted of several lectures, which consisted of text units. Several courses belonged to course group. A Content Map was built manually by the author of the information resource. For each HTML page, the content map was the set of knowledge items on page. Using these content maps, the HyperBook system dynamically computed links to related information. Related pages were those whose content maps were not disjoint to the maps of the current page. While this system did not

adapt to individuals, it did present a way of structuring more complex HyperBooks from distinct sources.

Dynamically adding links to a hyperspace can turn a static into a personalized one since the links are computed on an individual basis. However, since links are computed dynamically, the structure of the hyperspace may change while a user uses the system. This can be very confusing to users if they are expecting links from one node to another to always exist. With dynamic linking, that may not be the case.

3.1.2 Adaptive presentation of the material

While adaptive navigation helps students through a hypermedia space, it does not change the actual contents of each node in the space. Adaptive presentation, on the other hand, does change the content shown to users. In this section, we present some techniques used for adaptive presentation of the material. These techniques range from having conditions under which certain parts of a page are shown through full natural language generation techniques.

3.1.2.1 Adaptive presentation based on conditional text

Conditional text is a technique where parts of the page have conditions for their inclusion in the final displayed page. These conditions are evaluated when a page is to be displayed to determine if the conditional text should be included.

One of the first web-based systems to use conditional text was C Book [Kay, Kummerfeld, 1994]. In this system, each page was written in HTML, with parts enclosed in conditional rules. These conditional rules were processed and compared to the user model to determine if they applied. If so, the conditional text was presented to the user. The user model in this system included information about the user's knowledge of programming languages and how well she knew various concepts in those languages. It also included information about her preferences, such as abstract versus concrete, terse versus more detailed and explicit, and active versus directed. This was similar to the learning style preferences we are storing in BAHM.

The AHA! system [De Bra, Calvi, 1998] and its predecessor used an extremely similar approach for presenting web pages. Each page contained hard-coded information

and conditional information with if-then rules indicating if that information should be provided. These conditions of these rules were compared to the user model to determine which ones evaluated to true and which ones to false. If the rule evaluates to true, the text under the *then* part of the rule was included; otherwise, text under the *else* part of the rule (if it exists) was included. Saiz [Saiz, Szekely, and Devang, 1998] presented DPML (Data Presentation Markup Language) an extension to HTML which allow for personalization. DPML files contained meta-models for content presentations. For each request, the Presentation Agent (PA) read the user profile and then read the DPML file to create tree of objects that represent structure of document. For each object, the PA read the meta-model files and fetched the data. The PA then applied style rules. Presentation tailoring was done through style rules (condition/action pairs), conditional tags (whether whole fragment of HTML should be included), and cascading style sheets.

Conditional text is a very straightforward technique that is simple to author and to implement. The course designer must decide only on the instances for when a piece of text will be shown. But because it is very simple, it is not very flexible. Each conditional piece of text must have a rule associated with it, rather than a general rule to apply to an entire class of text. For example, say an author wants to indicate that if a student does not know a particular concept, examples on that concept should be shown. With conditional text, this general rule cannot be implemented. Rather, each example text on a page must have a rule surrounding that text indicating to show that text if the user does not understand the concept.

3.1.2.2 Alternatives

One technique of adaptive presentation is to have multiple ways to describe a piece of content. A user model is used to distinguish between these variants and to choose the correct ones.

One system that employed such a technique is ANATOM-TUTOR [Beaumont, 1998], which was an intelligent tutoring system for teaching anatomy that had a hypertext presentation component. The user model was based on stereotypes, but could be individualized. When a user first started the hypertext component, some basic text was chosen for her based on her stereotype, which was then individualized based on her

individual model. The basic text was divided into units of one several sentences. The units were arranged according to user levels and according to what information was in the unit. Text units could be deleted if they were assumed to be known, or added if they were needed. AVANTI [Fink, Kobsa, and Schreck, 1997] also used variants to customize the presentation of web pages AVANTI was a hypermedia system, which was to be used by different classes of users, including some with disabilities such as dystrophy. The idea was to adapt the content based on some stereotypes of users, as well as individual user models. Each page contained static elements and possibly optional elements. Adaptation Rules were used to determine which of these optional elements to include. These rules took into consideration the user's model. Furthermore, each page also contained rules on updating the user's model based on her interactions with the page.

The SmartGuide system [Gates, Lawhead, and Wilkins, 1998] used a variant on fragments for personalized content. The goal of this project was to customize the web interface so it matched a user's goal. The system could integrate resources from heterogeneous sources into a unified interface. Each page in the system was a template with a fixed part and dynamic parts. The dynamic parts had parameters on how objects (related data items) were to be selected. Pages were rendered by constructing one or more queries based on the parameters of the dynamic parts of the page and the user model, comparing the queries to information objects, and substituting the best matches into the template file. User models were stereotype based, but could be individualized through direct user feedback. The user also provided information on her short term goal, i.e. why she was using the system for this session.

The EPIAM system [De Rosis, De Carolis, and Pizzutilo, 1994] provided tailored explanations of concepts. In this system, concepts were classified by their theoretical meaning. Each class had a set of attributes and each attribute had two texts associated wit it: nucleus and satellite. A schema-based approach was used to generate messages. In this system, schemas defined which attributes should be described, and whether just the nucleus or both the nucleus and the satellite should be used. The system analyzed the user's model to determine which schema selection rule to apply. The user model was stereotype based and contains estimates of the user's knowledge on concepts. The stereotype was triggered by answers to general questions on the user's curriculum. However, the estimates of a user's knowledge could be updated, since the concepts in the user model were stored as a belief network. In case the generated explanation was not

sufficient, EPIAM supplied hypermedia follow-up which allowed the user to obtain mode detailed information.

APHID [Kettel, Thomson, and Greer, 2000] used a form of variants. This system used templates and patterns to adapt the content of a page. The patterns were used to determine which input elements from the document would be processed by the template. Rules were provided which map from the student model to the adaptations that could be performed. Some of the parameters that could be set on a page were the number of data elements per page, the maximum number of images, and the length of text on the page. The system presented in [Lin, Danielson, and Herrgott, 1996] performed page variants. The instructor module in this system selected which HTML page to send to the browser. By doing this, the instructor module could vary the level and the medium of the HTML documents, based on the student model.

Rousseau, et al.[Rousseau, Garcia, Valdeni de Lima, and Duda, 1999] presented, not an adaptive system, but an adaptable system using variants. In this system, each user specified a "predicate" to create a personalized view of document. A "predicate" include the user's end platform, her network connectivity, her accessibility, and her needs (user profile, education level, time, access history). The system designers focused on synchronized multimedia presentations with the goal of customization of a generic presentation to user needs, using temporal extensions to HTML. Temporal links related two media sample and assigned time instants to the samples. These temporal links and domain objects had alternate content, for adaptability, which could be selected based on user's predicate.

SETA [Ardissono, Goy, 2000] was a personalized shopping system which used stereotype user models that could be individualized. In this system, one agent selected the products for the user to view and another agent generated the customized pages for the store. User models contained domain independent information about a user's characteristics – her ability for absorbing large amounts of data and her requirements for web-page layout – her technical and aesthetic interest, and her expertise on various product classes. The model also contained domain dependent information defined for each web store. The system judged a user's interests and performances by observing the actions she performed, such as whether an item was added to the cart or if the user asks for more details. Once a product had been chosen, the details about that product were dynamically generated. The content was tailored to the user's interests and receptivity and the selection

of the linguistic form of the descriptions was tailored to the user's expertise. The layout of the page was also personalized to the user.

Another personalized shopping system was TELLIM [Joerding, 1999], which used only temporary user models – single session – to accomplish the personalization. The system attempted to learn the user's preferences for the kind of content displayed based solely on implicit feedback. The system used an incremental learning algorithm based on CDLA. The attributes were type of medium, kind of product, brand, and kind of information. There were rules in the system to update the user model to determine which content to show.

[Bental, Cawsey, Pearson, and Jones, 2000] described a system which used a form of variants to present individualized information on cancer treatments. The domain was organized by issues and text about those issues. Issues were selected based on what was relevant to the patient's treatment, specific illness, and current time frame. If an issue was selected by the patient, texts about that issue were selected, based on both the patient's record and on annotations about the text that provide details of conditions when the text is appropriate. Text plans associated with issues indicated which texts to select and how they should be grouped to provide a coherent presentation. By using the variants technique, a system can present essentially the same information in a personalized way to a variety of users. These systems can therefore be very general and flexible, using general rules that cannot be used with conditional text. However, they can also be very intensive in terms of domain construction; somebody must construct the variants and rules for presenting them. Furthermore, if the system is not correct in its assessment of users, it may present the wrong variant. In these cases, system must provide the means for users to access the other variants.

3.1.2.3 Stretchtext technique

Stretchtext is a technique that replaces a word or a key phrase with additional information on that phrase. Whether this additional information is shown, and what kind of information to show, is decided on an individual basis. This is the technique we are using in BAHM.

MetaDoc [Boyle, Encarnacion, 1998] is a system that used stretchtext. This system used four stereotyped user classes. Concepts that were stretched were classified using the

same scale. If a user was presented with a concept with which she was not familiar, additional explanations were included, but they were not included for lower level concepts. Furthermore, higher level details were not provided for lower level users, but details of lower level concepts were displayed for higher level users. The system used both explicit and implicit user modeling. For the implicit modeling, the system evaluated the kinds of requests made by the user. Requesting more explanation about a concept implied a lack of familiarity, while requesting more details implied an understanding. Similarly, requesting less explanation implied an understanding, and requesting more details implied unfamiliarity.

KN-AHS [Kobsa, Nill, and Fink, 1997] was an adaptive hypermedia system that does hot-word adaptation. Additional information about the hot-word was presented and customized based on the BGP-MS user modeling system. This user modeling system used both stereotype user models and individual user models. The rules of adaptation were:

1- If the user was unfamiliar with the hot-word, an explanation was automatically added and an icon for a graphic provided.
2- If the user was familiar with the hot-word, more details were automatically added.
3- If there was no information about the user's familiarity, the hot-word was not changed.

There were corresponding rules to determine if a user was familiar of unfamiliar with a hot-word:

1- If the user requested an explanation, a graphic example, or a glossary definition, then she was not familiar with the hot-word.
2- If the user unselected these, then she was familiar.
3- If she requested additional details, then she was familiar with the hot-word.

The ADAPTS system [Brusilovsky, Cooper, 1999] also used a stretchtext interface. Paragraphs in the troubleshooting procedure were expanded or collapsed, based on the user's knowledge of the task or if the information contained in the paragraph was not relevant to the current context.

The PUSH [Höök, Karlgren, Waern, Dahlbäck, Jansson, Karlgren, and Lemaire, 1996] also applied a form of stretchtext for adapting its presentation. In this system, all valid data about an object was offered in one page, with some of the objects possibly being hidden. The system determined, based on the user's data looking for task, which information to show and which to hide.

3.1.2.4 Adapting everything else but the text

The systems discussed so far in this section primarily do adaptive text presentation. However, there are other aspects of hypermedia that can be adapted to individuals. One application of adaptive presentation is adaptive quizzes. In [Pérez, Gutiérrez, Lopistéguy, 1995], the authors discussed just such an application. In this system, HyperTutor, exercises were dynamically generated to include questions on concepts recently seen. Items were selected based on rules, such as multi-concept items were preferable to single concept items, items on which the student had previously failed should be presented, and items testing already known concepts should not be presented. Exercises were generated either when the student asked for them, when the hyperspace contained an exercise node, or the system decided to present an exercise.

Another system which preformed automatic exercise generation was Multibook [Fischer, Steinmetz, 2000]. In this system, the goal was the automatic generation of simple exercises, i.e. generation of multiple choice questions and distractors. The domain model was constructed in such a way as to support *part-of questions*, which ask about the parts of concept. It also supported *is an application* for questions. For both types of questions, the system dynamically constructed a list of distractors and a list of correct answers, randomized the order, and checked to see if the user selected the correct answers. If wrong answers were selected, the system offered to branch to explanations of wrong concepts and to provide a repetition of the lesson being learned.

TANGOW [Carro, Pulido, Rodriguez, 1999] used adaptive presentation techniques to generate individual pages, selecting which type of media elements would appear in the document and how they were laid out based on the student profile. Media elements were selected based on features such as content difficulty and the language in which they were written. In InterBook [Brusilovsky, Schwarz, 1997], the interface to the system was adapted to each user. In this system, the adaptive presentation was accomplished by incrementally adding interface features – such as a glossary button – to the HyperBook layout whenever the student was ready.

The aim of the BAHM [Aïmeur, Salehian 2003] is to build a web-based environment which not only is adaptive to the user, but also gives the ability to the user to learn by doing. In BAHM, concepts are related to each other on the base of a conceptual

model of the hypermedia-type systems. Observations about users are made when a user has performed some action inside the virtual environment. Each unit is indexed with some knowledge concepts. A separate knowledge model is constructed, containing the knowledge concepts of the application domain and their learning dependencies. Thus, the documents itself do not contain any prerequisite or outcome information. A glossary containing the concepts of the knowledge model is generated. For each glossary item, links to examples, to material units, and practice environments are generated.

BAHM used the dynamic links for Link level adaptation. A page sequencing algorithm generates reading sequences according to the user's goals and knowledge. For helping the user to find her way through the course material, the system also generates a next learning step for the user by comparing her actual knowledge state with knowledge he should have after finishing the unit.

BAHM supported explicitly goal-based learning. Users could define their own learning goals or could request the next learning goal from the system. For each of these goals, a reading sequence containing necessary knowledge (prerequisite and actual necessary knowledge) for reaching the goal is generated. In addition, having tree view navigation gives more flexibility to the user to select the next goal.

BAHM also adapts to the different learning speeds of the users by supporting this kind of goal oriented learning. Users can define how much and what to learn next. If the system observes a user mastering an advanced action sufficiently well then it updates its estimation about this user in relation to the mastered topics as well as to prerequisite knowledge concepts. Thus the user can go on with further, advanced topics. If the system observes that a user is not quite familiar with some topic, it proposes similar examples or actions, which contain only a few amount of new information.

3.2 Discussion

Some of the systems discussed so far in this section primarily do adaptive text presentation. However, there are other aspects of systems that can be adapted to individuals. One application of adaptive presentation is adaptive tests [Aïmeur, Salehian, 2003]. In [Pérez, Gutiérrez, and Lopistéguy 1995], the authors discuss just such an application. In this system, HyperTutor, exercises were dynamically on rules, such as

multi-concept items were preferable to single concept items, items on which the student had previously failed should be presented, and items testing already known concepts should not be presented. Exercises were generated either when the student asked for them, when the hyperspace contained an "exercise" node, or the system decided to present an exercise.

Another system which performed automatic exercise generation was Multbook [Fischer, Steinmetz 2000]. In this system, the goal was the automatic generation of simple exercises, i.e. generation of multiple choice questions and distractions. The domain model was constructed in such a way as to support part-of questions, which ask about the parts of a concept. For both types of questions, the system dynamically constructed a list of distracters and a list of correct answers, randomized the order, and checked to see if the user selected the correct answers. If wrong answers were selected, the system offered to branch to explanations of wrong concepts and to provide a repetition of the lesson being learned.

We conclude from these experiments that a standard strategy for all students clearly does not lead to maximum performance. Therefore, systems must provide different strategies and must have methods for determining which strategies are most appropriate for each user.

Also from the results in this section, we have chosen not to develop an standalone ITS, but create a Web-based tool – BAHM – with link annotations, dynamic link and direct guidance for the adaptive navigation – a tool which embeds a virtual environment to provide users with ability to learn-by-doing (see the chapter VI). We can simply notice that the virtual environment underlying the BAHM is more useful than the techniques in the other hypermedia like approaches, since it practically involves the users in a real world problem. In the chapter 4 we discuss the teaching methods and the theory of learning style which applies to BAHM's virtual learning environment.

Chapter IV

4.Tutoring the Learner

Since hypermedia systems and some of the multimedia systems are web applications, they are typically used in distance learning scenarios, where a learner uses the data from the system on her own. Thus, it is essential to think about useful teaching strategies to persuade a learner to learn actively and not passively. For this purpose, we emphasize Kolb's learning strategies, for instance by put together problems/real tasks in the domain model of the system, and by constructing the environment based on scheme. Users can achieve learning goals as well as receiving answers to data requests while working on the problems, which brings up explanation of the learning items [Henze, Nejdl, and Wolpers, 1999].

BAHM's primary goal is to help *SoftPhone* users understand all the features of the software and get the best use of the software. This idea started when we noticed that not all the features of the software are recognized and used by the users. As an example, MS Word which is a well-known software developed by Microsoft has been used by almost all the students. Considering the money spent on developing this software and creating great abilities, but not everybody is aware of all the MS Word's abilities and the company had to spend a huge budget to create an on-line help to teach this software to its users. Besides the money-wise problem this kind of help – on-line help – are not always helpful, because the contents of the help are static for all the users and this would cause the finding an appropriate help troublesome.

BAHM is designed to be used by *SoftPhone* telephony users via the Internet. Because on one hand, installation and configuration of the software are often complex, virtually all the users need some assistance beyond the instructions and reading of the manuals. Even when the users have learned the facts, they need help to apply them in

problem solving. On the other hand, users of the system are located in different cities and countries and this is impossible for the company to send an expert to the remote locations for installation, training or troubleshooting. Therefore, BAHM's domain provides an ideal environment for the study of tutoring and student modeling.

4.1 Specifications of the Distance Learning

Computational learning environments take advantage from a strong background in educational theory [Roberts, Pane, Stehlik, Carrasquel 1988]. Reproducing predictable teaching and learning concepts in a computational environment does not utilize these new technologies. Educational models, which provide mainly interesting features for many parts of academic education, are constructivist models of learning and teaching.

If Virtual learning environments considered appropriately, they may offer the functionality to support improved concepts in education theory, which are difficult to understand without the help of new communication – and networking technologies. Critical fundamentals in the design of such learning environments are the requirement and incorporation of authentic and complex activities during the learning process.

In a virtual environment user suppose to solve a problem as if it is occurring in the real world. In fact the learning environment simulates the context of the problems, on which the user executes the authentic actions: they decide how to and solve the problem, using their experiences. By that means responsibility for both selecting and performing tasks moves from the teacher to the learner. This is the way that the users get actively involved in covering the course material. Obviously, the teacher's role is to prepare the student's work and to teach them the preliminary essential concepts for their task. Of course real-world examples are not that simple and usually are more complex in which they arise. To present abstracted exercise problems to students, should employ simplified forms of techniques and applications. However, abstraction is necessary and small exercises can be used to argue specific issues, project-based learning has to be used to rebuild real-world complexity. Two things should be pointed out here; global project context establishes the learners' outlook on a given task, and subtasks present guidance of the learning process.

Collaborative learning extends the ability of analyzing the problem from numerous views. Keeping in mind that project work is often done in teams, learners can train their capabilities for team-work and collaboration.

4.2 Theory of Learning Style

This section presents different definitions of cognitive styles and learning styles, these two definitions are sometimes misinterpreted and are used vice versa. Our focus here is more on learning style than cognitive styles. Learning style is about specific personalized approach of each learner to new knowledge achievement [Cristea, and Okamoto, 2001]. Based on the student's learning style, the learning style is autonomous from the other skills and each student has different ability to receive the learning material via a certain teaching style. Moreover, learning styles are mental behaviors that serve and steady signs of how learners perceive, and respond to the learning environment. Another words, learning style is the way learner absorb, process, and maintain information. All theorists view learning styles in exactly the same terms, yet their methods for judgment and interpretation may differ.

The education literature proposes that students who are actively involved in the learning process will be more likely to achieve success [Kuhn, 1972]. A key to getting students aggressively concerned in learning lies in understanding learning style preferences, which can positively or negatively affect a student's performance [McCann, 1975]. It has also been revealed that fine-tuning teaching materials to meet the needs of a variety of learning styles benefits all students [Agogino, Hsi 1995].

[Birkey & Rodman 1995] show that, just as there are "remarkable differences in the way people learn and process information, there are significant differences in how learning styles are defined and measured." Among the different cognitive, learning styles, we are enumerating some of the more important in the following [Cristea, De Bra, 2002]. We based our thesis on one of them – Kolb's theory.

4.2.1 Hill's Cognitive Style Mapping[5]

Joseph Hill was among the earliest theorists in the field of learning style [Hill 1981]. He described learning style as the sole way in which an individual searches for meaning. Hill believes that process was revealed in three following categories:

- The processing of theoretical and qualitative symbols
- Modalities of inference
- Cultural determinants

He categorized the theoretical symbols into auditory and visual. Each one is subdivided into linguistic and quantitative symbols. Moreover, there are fifteen qualitative elements. Among these fifteen, there are empathy, proxemics, which is social distance, proprioceptivity, and a sixth sense [DeBello, 1990]. Hill believes in modalities of inference as the second major category. There is the variety of conjecture an individual uses in the process of actually attaining meaning. Majority of elements in this category are as follows: critical thinking, contrasting and comparisons, relationships between measures, and hypothesis development, some of them are redolent of Bloom's taxonomy [Hill 1981].

Hill introduces the third major element of his model as cultural determinants. Hill saw how persons understand symbols, and he believed that the meaning that is assigned to symbols is shaped by one's culture [DeBello, 1990]. Based on Hill cognitive style mapping is the fairly multifaceted procedure of attaining a cognitive style profile.

4.2.2 Kolb's Theory of Learning Styles

Dr. David Kolb refers to this framework as an experiential learning model. The center of his model is an explanation of the learning cycle [figure 4.1]. His consideration is on how an adult's experience can be translated into concepts, which ends up in the choice of new

[5] taken from DeBello.

experiences. Kolb's theory is based on a cycle which explained as follows [DeBello, 1990]:

- Immediate concrete experience is the basis for observation and reflection.
- Observations are assimilated into theory from which new implications for action can be deduced.
- Implications serve as guides in acting to create new experiences.

In order to be effective, the learner requires abilities that are opposites such as:

- Concrete experience versus abstract conceptualization
- Active experimentation versus reflective observation

However, as the outcome of experiences, and the demands of present environments, the majority of people develop learning styles that emphasize certain learning abilities over others.

Kolb's Theory of Learning Style is a nine-item measurement with four sub-items to be ordered by adults. Four leading types of learning styles have appeared most frequently [DeBello, 1990]. First, *Converger,* whose dominant learning abilities are abstract conceptualization and active experimentation. Most advantages of this type are in IQ tests, decision-making knowledge acquisition [Cristea, De Bra, 2002]. The second type is *divergers,* who have opposite learning potential of the converger. The supreme strength of these persons lies in their creative skill. They take pleasure in viewing concrete situations from many outlooks. The third one is *assimilators* who learn via abstract conceptualization and reflective observation. Their ability is to create theoretical models.

The fourth type is *accommodators* who learn via concrete experience and active experimentation. Their potency lies in doing things, in running plans and experiments, and involving themselves in new experiences.

Kolb [Kolb 1981] illustrated that learning styles might be seen on a variety running from (see figure 4.1):

- Concrete experience: getting involved in a new practice
- Reflective observation: observing others or developing observations about own experience
- Abstract conceptualization: generating theories to explain observations
- Active experimentation: using theories to solve problems, make decisions

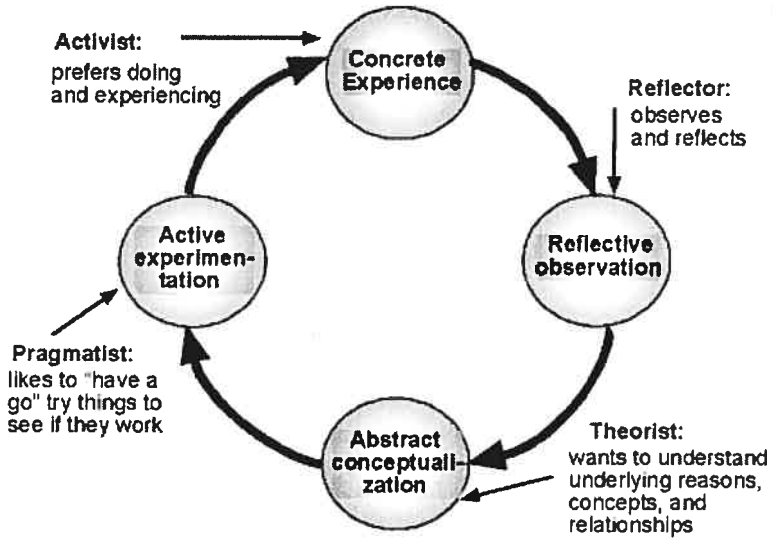


Figure 4.1 Kolb's continuum running cycle

Hartman [Hartman, Virginia 1995] obtained Kolb's theory of learning styles and proposed examples of how one might teach to each of them:

- For the concrete experience – propose laboratories, interpretations or activate films
- For the reflective observer – use logs, diaries
- For the abstract conceptualize – lectures, papers
- For the active experimenter – offer simulations, case studies and training

Kolb's model and instrument were designed for and have been applied to adult organizational systems and management training. At least four different variations of Kolb's model are in use today [DeBello, 1990].

In BAHM, we created an environment in which users can be involved in a new experience by passing a pre-test. This test will be done inside a virtual environment. She can ask for the help to observe others and compare her solution with system's solution.

4.2.3 Dunn's Learning Styles

Rita and Kenneth Dunn developed a comprehensive learning style model on four axes in 1974 [Cristea, De Bra, 2002]:

- *Environmental factors* (noise, light, sound, temperature, design setting)
- *Emotional factors* (enthusiasm, determination, accountability, formation)
- *Sociological factors* (self-orientation, authority orientation, coworker orientation, group orientation)
- *Physical factors* (insight, transportability, era)

Even though this model deals very little with the cognitive factor, this model is currently used in schools for pupils of grades 3-12 and a version has been developed for adults.

4.2.4 Learning style preferences and adult Learning Styles

We should know is that what makes adult learn differently from kids do. Theory of andragogy – the art and science of helping adults to learn – is a shot to tell apart the way adults learns from the way children learn. A number of assumptions are made based on Cantor [Cantor, Jeffrey 1992]:

- Adults are independent and self-sufficient.
- Adults are objective oriented.
- Adults are relevancy oriented
- Adults are realistic and problem-solvers.
- Adults have built up life experiences.

Learning styles is "the ways in which different people think and learn." [Litzinger, Osif 1993]. Every one of us creates an ideal and owns a set of behaviors or approaches to learning. Litzinger and Osif summarize the learning procedure as several steps:

1. Cognition, they way a person obtains knowledge
2. Conceptualization, how a person analyzes information.

3. Affective, emotional preferences which help to define the learning styles.

4.2.5 What Motivates Adult learners?

Adults in general, have different enthusiasms for learning than children do, such as those pointed out by Cantor [Cantor, Jeffrey 1992]:

- To make or maintain social relationships
- To gather outer potentials
- Improve serving others
- Expert development
- Escape or motivation
- Net interest

Considering the fact of motivation for learning, instructors should pay more attention to the possible motivations behind their students' enrolment. Then they can better shape the instructional materials. This consideration should be in mind at the time of developing an ITS too.

4.3 Excursion: The BAHM Virtual Environment

One main goal of developing BAHM has been to utilize the full power of virtual learning environment techniques to innovate teaching and learning in our courses, instead of just transplanting ordinary lectures onto the Internet [Salehian, Aïmeur 2003]. We have used the theory of learning style proposed by Kolb to create the BAHM's VE and we got benefits of specifications of the distance learning to make the BAHM accessible via Internet at remote locations.

The BAHM virtual environment provides the environment for using adaptive hypermedia to teach the users how to configure their system, how to install the *SoftPhone* telephony, how to make the best use of the product in

the shortest time, and more importantly how to deal with problems that arise. In the first stage of the development, BAHM has been evaluated and the results from the evaluations were directly used by every subsequent implementation of both the virtual learning environment and the adaptive hypermedia part of the BAHM system.

4.3.1 Availability of the Working Environment

In order to make the working environment continuously available, access to all parts of the working environment (including all information and software tools) are enabled via the internet. Most tools are available for all current operating systems or browsers, so users can use them locally at every computer they have access to, all around the world.

If license restrictions make this impossible (as in the case of a large software engineering tools), at least anytime access over the Internet is available. Locally used tools are always internet-based, so access to central servers, repository, and communication facilities is always possible. Most of the on-line information is also downloadable for off-line use.

4.3.2 Electronic Communication Facilities

Each user has a communication center, which includes e-mail address of the users within this center as well as user name to enter. The system also includes a conference room which enables the users to communicate and share the information with each other (see figure 4.2). There is always one *Expert* available in the conference room for the urgent help. In case that users face to a problem on which they need more help, they can communicate with the *Expert* via this conference room

The messages and discussion forum are also automatically archived, indexed, and made available over the Internet. Synchronous communication at present can take place over a text based conference room, the *BAHM on-line Conference Room*. All facilities are available on many operating systems, and on computers connected to the Internet anywhere.

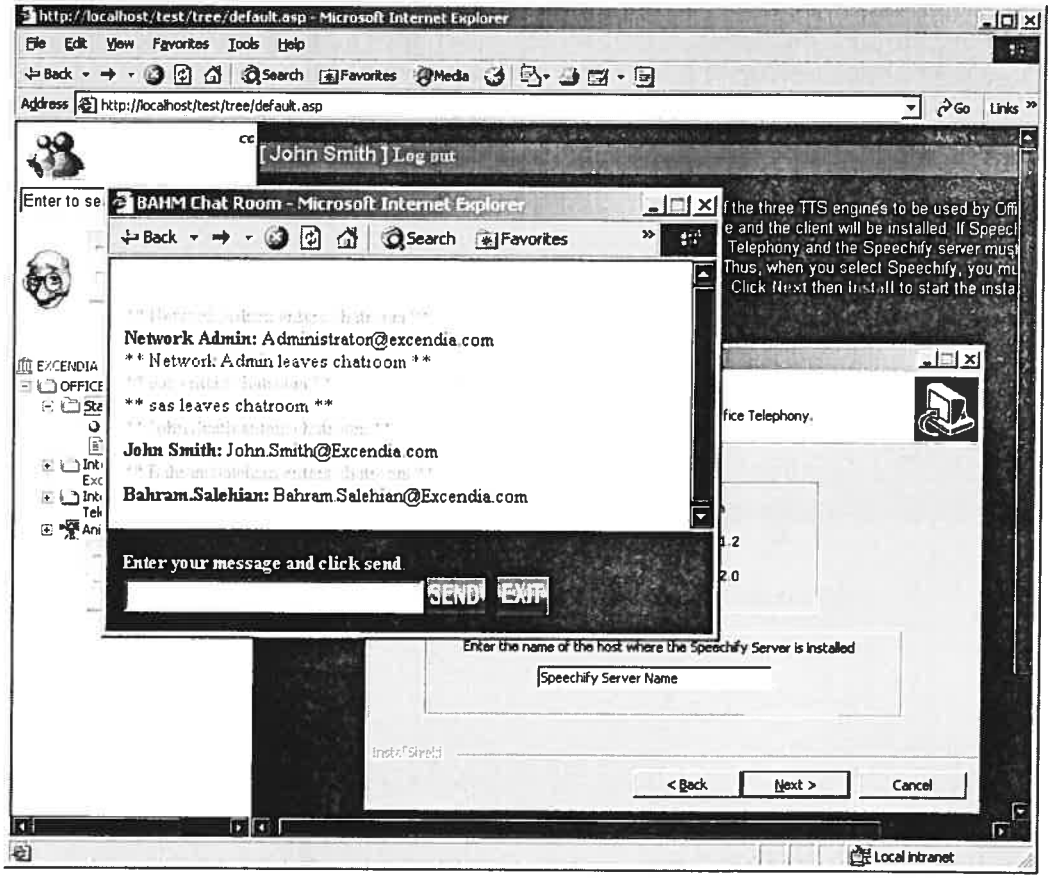


Figure 4.2 Snapshot of the Conference room in BAHM system

4.3.3 Network Environment

A server located at Excendia Company as the central repository for all information. Such as topics, sub-topics, Video animation, web pages, interactive multimedia related to each project, etc.

All central repositories (WWW communication and presentation area, etc.) are also stored at the Excendia's server. A variety of working environments for accessing the data is supported.

4.4 Discussion

It is a central requirement of tutoring approach in the BAHM to keep the system as user-friendly and comprehensible as possible and keep the student interested and motivated. Based on Kolb the first step to teach a student is motivation and then self-reliance and self-confidence. A user-friendly structure allows the students to understand how to get the best of the system without help. Since BAHM is Web-based system, self-reliance and self-confidence will be a major point for the student to follow the course.

Another important requirement is the information presentation in this learning approach. Since the students are supposed to work practically on their own, therefore it is a task of the system to select and present suitable information as well as appropriate tests to a user. Since the students are working with BAHM in the Internet, the integration of useful information present in the Internet into the learning material is near at hand.

To reach such a (proposed or self defined) learning goal, the system should be able to find relevant projects and examples related to the goal and allow the user to practice on the project inside the virtual environment. Therefore, selecting algorithms have to be found. They should present the most suitable projects to the user which match to her current learning goal and consider her actual knowledge.

The BAHM system implements the above stated requirements. In chapter 5 we describe the architecture of the BAHM system as well as its functionalities.

Chapter V

5. The BAHM System and its Architecture

The Bimodal Adaptive Hypermedia and interactive Multimedia – BAHM – is a tool for modeling, organizing, configuring, troubleshooting and maintaining the telephony software on the WWW. The system is developed at the *Excendia* Incorporation. In this chapter, we discuss the details of the BAHM system as well as its architecture. We also discuss why this architecture was chosen, instead of other architectures that are used with other Web-based systems.

5.1 Objective

The objective of this venture is to contribute to the enhancement of the learning process by developing a highly developed training system for software maintenance training, based on novel concepts, new cognitive approaches and simulation technologies. The improvements of the learning process refer to:

- reduced time, costs and risks of training
- improved features to deliver complex contents to trainees
- more *learning-by-doing*
- more attractive training
- the possibility to learn by mistakes

- improved means to provide a more general understanding of the technical system as well as relationships and interdependencies
- provision of means not only to raise the awareness of wrong configuration issues but by experiencing them personally

Software developer organizations are faced with a number of challenges that BAHM is designed to address

- Software are becoming increasingly technologically complex, making the technical competence of technicians even more critical
- A world-wide shortage of maintenance technicians that is forecasted to accelerate, increasing demands for cost effective training. This also increases the demand for quality on the job training to reduce the time from initial recruitment to productive participation.
- Competition from Asia and third world maintenance organizations in cheaper labor markets has meant that European and North American maintenance organizations have had to rationalize operations and find ways to compete (such as on quality). BAHM helps to reduce the cost of investment in personnel.

5.2 The architecture

The architecture of BAHM consists of two main parts (see figure 5.1):

- The server side
- The client side

In turn each side contains few parts as follows:

The server side contains:

- The Student model server (the highest layer from above in figure 5.1)
- The planner
- The Session manager

And the client side contains:

- The Panel updater
- The Helper
- The GUI (Graphical User Interface)

The entire processing and "intelligent" decision making occurs on the server side of the architecture. All a student needs to run a BAHM course on the client side is a web browser (GUI in the figure), RealOne, and Flash player which runs the virtual environment of the system.

The browser window of the BAHM course material contains two frames (see figure 5.2). The right frame displays the HTML, and ASP (Active Server Page) course materials as well as the Multimedia type virtual environment in which user can learn by doing. The left frame contains a button for entering into the Conference Room, an input field which is connected to a search engine, an agent to answer to the student's questions as well as a tree view navigation tool to allow student to see the table of contents, and navigate forwards or backwards through the material.

On the server side, the HTTP server interacts with port server and with student model server which contains the data about the student. The port server controls the creation of the student model and one student model server is created for each student using the system.

The main communication link is between the HTTP server and the student model server. Each learner is assigned a unique "*Session ID*" at the time of login by the Session manager and the HTTP server uses the "Session ID" to maintain state with a given client, and based on this Session ID determines which student model server to contact on each interaction. The Session ID will remain in the system until the student logs off.

When the HTTP server is contacted, the correct script is invoked and a Session ID is assigned to the student. The ASP controller reads what is in the query string, and passes this information on to the correct student model server.

The *Planner* is responsible for all the reasoning and dynamic construction of course content. The Planner continuously runs between student model server and Session Manager, thus maintaining the state. This is one of the reasons that this architecture was chosen; we did not want to have to rebuild state each time the HTTP server was contacted. The Session Manager and Planner are responsible for logging all the student actions for the student model and for generating the content the student sees as well as communicating with the Helper to provide assistance in case of necessity.

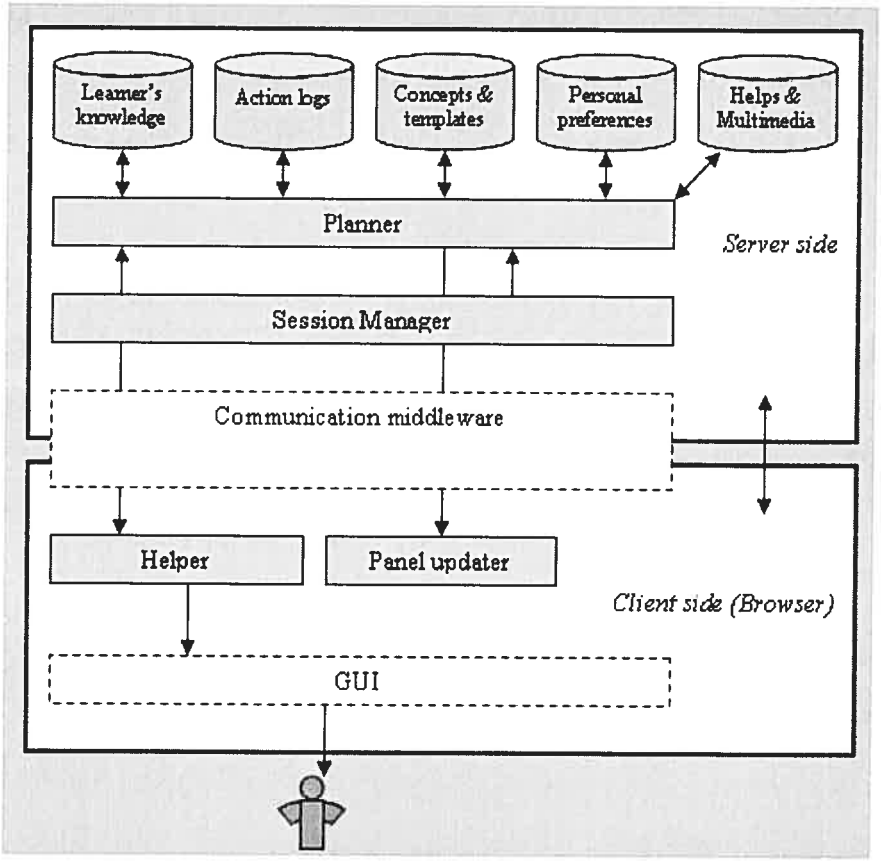


Figure 5.1 System architecture

5.3 Why we chose this architecture?

Many different architectures exist for the web-based educational systems. Some of the very first network based educational systems used specially created client applications supplemented with a standard HTTP browser [Lin, Danielson, Herrgott, 1996]. Others use specialized HTTP servers that can be modified to incorporate teaching components [Eliot, Neiman, LaMar, 1998]. Still others use a combination of client side applets and specialized servers. Most of these architectures did not meet our requirements. For

example, we need to record and remember every student action. We also wanted students to be able to learn a course through multiple sessions, possibly using different client computers and learn by doing. Therefore, we needed to be able to store the interactions in a student record that could be accessed multiple times from multiple locations. These requirements led to a server-centered design, with all the information being stored on the server and none of it on the client.

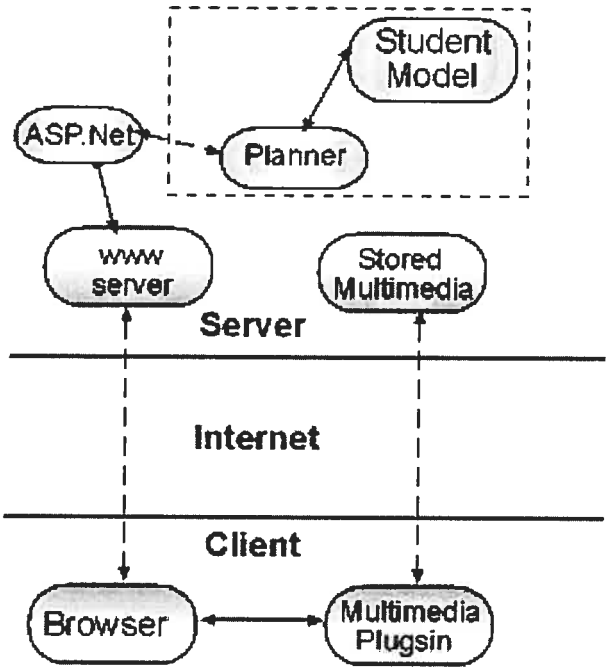


Figure 5.2 System architecture in the simple way

We also wanted an architecture that did not require special proprietary software. We wanted to use off-the-shelf components so everyone could learn by doing when they are using the BAHM system. We did not see the need to write the Java applet to capture student actions, since whenever the course is not extremely interactive a Java applet would still have to report the student actions to the server to be stored. Furthermore, we did not want to deal with rendering HTML code through Java applet. We rather have a browser do this for us.

5.4 Methodology

BAHM personalizes information according to the user's needs and knowledge. The system models some course, guides the user through the course and its learning material, and supports the user's access to useful information. The simulation-based teaching approach led us to create an adaptive system which enables users to learn by doing.

Didactic besides technical aspects of innovation, there are others e.g. in the area of instructional science. The merits of customized or tailored training for individual trainees with different learning styles and experiences have been realized since the early 1980s. Traditional computer-based training systems have relied on behavioral psychology emphasizing the analysis of jobs into a set of core skills. Components of skills could be mastered by following a prescribed and rigid sequence of exercises. The lack of opportunities to adapt the training exercises to the different learning needs of trainees and difficulty in improving the theoretical knowledge of trainees have led to further developments.

The literature of ITS has pointed out the need for developing computer-based training systems that could employ "learner models" in order to customize interactions to different stages in mastering skills for different individuals. Common criticism of many ITS include: the large amount of effort in preparing the training system, the difficulty of the human instructor to co-ordinate with the system author, the simplicity of application domains, and mainly the lack of empirical data to demonstrate significant differences with more traditional forms of computer-based training. One of the main factors of success of these training systems was the teaching of mental models, that is, practical theoretical knowledge which could enable trainees to transfer or generalize their skills from one situation to another. Earlier work in the area of qualitative or mental models of equipment had provided a useful basis to explore the issue of transfer of skills. The issue of transfer of skills, therefore, is very important for responding to the need of the industry to upgrade to ever increasing sophisticated machinery. Another lesson learned from the evaluation of ITS is the need to employ simulated models of devices or machinery in order to facilitate instructional strategies such as learning by doing and guided discovery. Simulated models of equipment offer excellent opportunities for acquiring skills such as "how-the-system-

works" and "how-to-do-the-job". A major advantage of simulation is that trainees are able not only to acquire a skill quickly but also to retain their skills in the long run.

This leads us to consider three important criteria for developing and evaluating new computer-based training systems. That is, acquisition of skills, transfer of skills to new but similar equipment and retention of skills even when practice on the job is not frequent. Virtual Reality (VR) is a computer-based training [Schreiner 2001]. With VR it is possible for the trainees to gain access to more realistic representations of the equipment they are likely to encounter in their daily jobs. It is anticipated that VR will enhance human memory of system components and required skills and thus, knowledge will better retained and transferred. However, applications of VR to computer-based training are lacking and there is a growing need for investing in such research developments. Our proposed system (BAHM) integrates VE (Virtual Environment) with various pedagogic issues such as training modes and competence measures.

5.4.1 Learning by doing

Since we based our research and project on Kolb's theory of learning style, we mostly emphasize on learning by doing.

- Where the trainee masters a skill by actively troubleshooting and solving problem simulated in the BAHM environment
- Guided discovery or "scaffolding" where the trainee undertakes a complete exercise even from the early stages of learning with the support of the system.
- Part-task training, that is, mastering component skills and then practicing them as a whole
- Heuristic training, where the trainee masters a common body of heuristic rules that help her narrows down possible software faults.

We will provide BAHM with a variety of measures for assessing different aspects of competence. Specifically, BAHM would address the following measures of competence or achievement:

- average time to troubleshoot

- accuracy in troubleshooting, i.e. how many faults have been identified in a set of exercises referring to the documented procedure
- Economy of strategy which reflects the number of information sources consulted in order to identify a device fault. Redundant checks, for instance, could indicate troubleshooting strategies which take more time to implement.
- Errors of omission and commission in exercises which present trainees with a "grey" fault symptom compatible with more than one fault.
- Recovery of errors committed earlier on in the learning sequence. This is an important aspect of competence in complex devices where errors in troubleshooting are bound to occur.

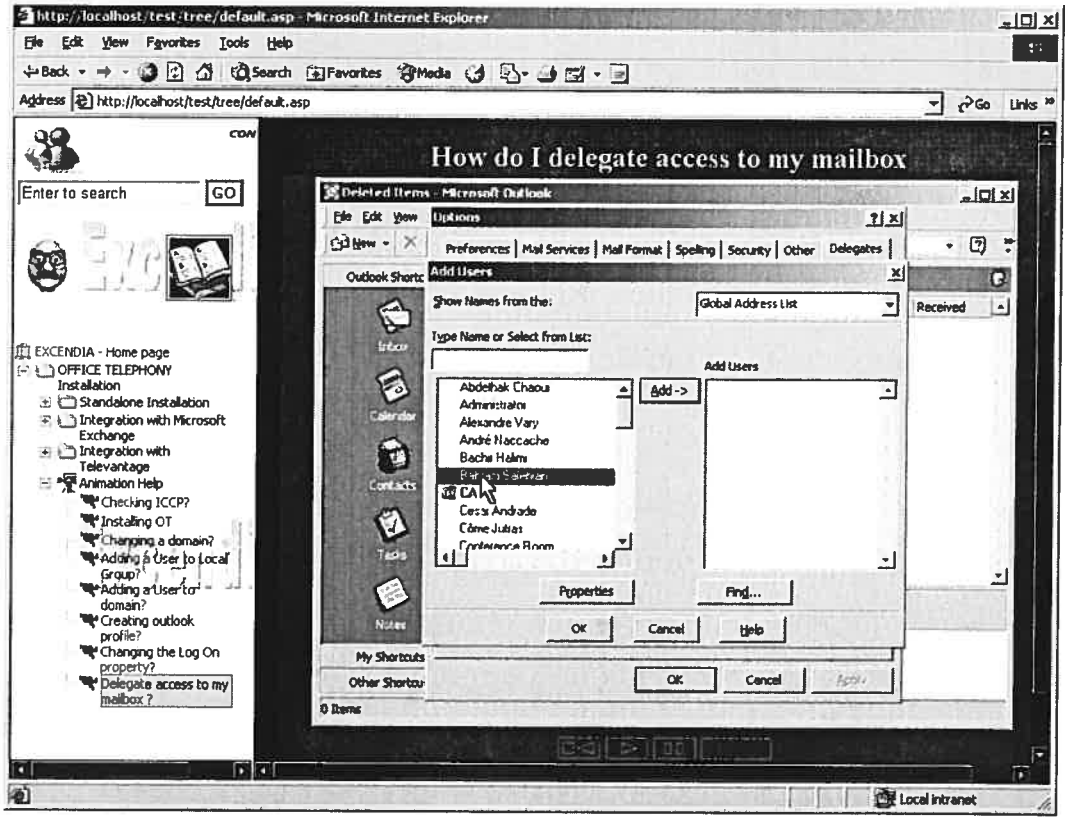


Figure 5.3 Snapshot of the BAHM's VE (left panel)

5.5 Modeling the Knowledge Domain

The main components of the domain are content objects (the actual content the student sees), *topics* (e.g. connection-oriented service), *concepts* (keywords that appear throughout the course, e.g. "PBX[6]", "ICCP[7]"), and test questions.

5.5.1 Content objects

The smallest unit of instructional material in the domain is the *Content Object (CO)*. In the current implementation of the BAHM, content objects are pieces of HTML and ASP pages, projects, and video help that are connected together to make a coherent presentation. There is no restriction to these media types; simulations, and flash presentation. Each content object has basic information which is always presented to a student. However, this information can be adapted, allowing students to see differing views of the course material. Other "features" of content objects are necessary in order to reason about the objects. The features for this BAHM courses are given in table 5.3 and describe the kind of content the object contains. The system uses these features to reason about the content without having to understand explicitly the meaning of the content.

Table 5-1 Features of the BAHM system

Features	Values
Media types	{Graphics, Video animation, interactive multimedia}
Instructional type	{Explanation, Example, Description, Definition}
Abstractness	{Abstract, Concrete}

Graphic objects:

The basic information associated with each graphic object is the file name containing the picture to be displayed.

[6] ***Private branch exchange,*** a private telephone network used within an enterprise.

Text objects:

A text object's main component is the text itself. A line of text can have subtext, which is one level in, and next text, which is on the same level.

Multimedia objects:

A multimedia object contains two parts: Video animation, and interactive multimedia. The video animation object is the file name containing the movies to be played. And interactive multimedia object again is a file name that can be called inside the ASP.Net page. Each one of these objects concerns a specific topic to be taught or to be tested.

5.5.2 Topics

The largest unit of instruction in the domain is the topic. We are using a standard semantic network to represent topics and relationships where labels on links include prerequisite, co-requisite, and related. This semantic network allows the topics to be linked together in a non-liner fashion (figure 5.4).

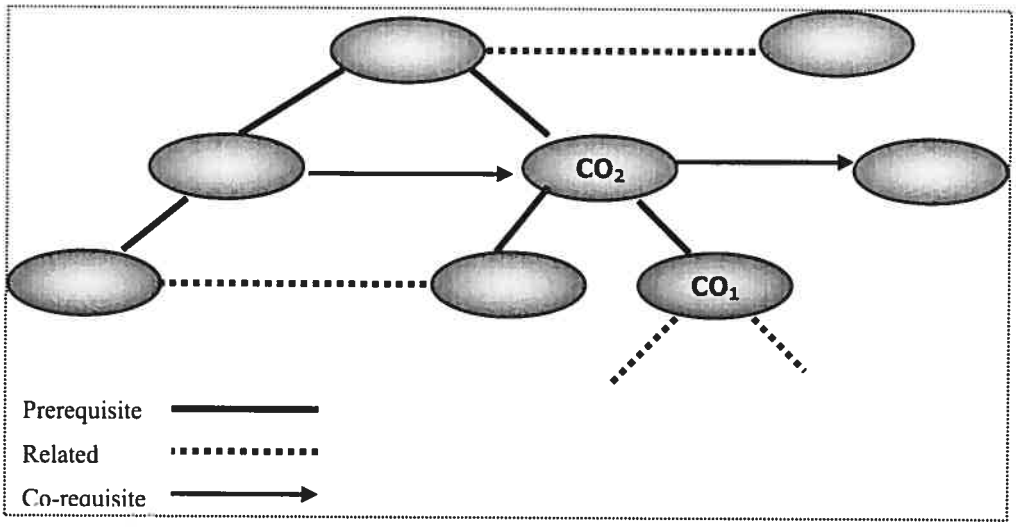


Figure 5.4 Link-type of the Content Objects

[7] Intel® Converged Communications Platform, standard-based building blocks for converged network interfaces and voice processing.

Furthermore, topics can have subtopics, which are, themselves topics. Thus the structure is recursive, and the topic tree can be quite deep (figure 5.5).

The connection between the BAHM system and the user modeling component is based on indexing any kind of information resources in the system. Here is an example to make it clearer.

Example: Concept Objects (*CO*) in the BAHM are, for example, *iccp*, *PBX*, *TeleVantage*[8], *standalone installation*, *MS Exchange configuration*, etc. (See appendix B)

Each topic has as many as three constituent parts: (1) the beginning material to teach (*before-objects*), (2) the subtopics, and (3) the ending material to teach (*after-objects*). Each of these parts does not have to be present for each topic. For instance, topics are not required to have subtopics. The beginning material and the ending material consist of content objects (figure 5.5).

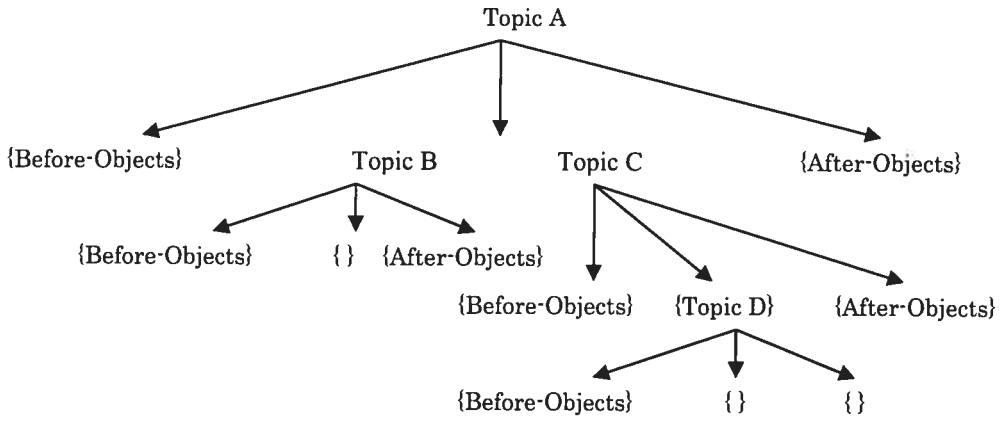


Figure 5.5 Topic structure

When presenting a topic, first the beginning material is taught, the subtopics are recursively taught in order, and finally the after-material is presented (this amounts to an in-order traversal of the topic tree). This topic structure was chosen to enable a course

[8] TeleVantage is the feature-rich, software-based phone system that combines the power of the desktop computer with the most advanced communications technology available.

constructor to have an introductory material and summary material surrounding a number of subtopics. It also allows a guaranteed page break, since when constructing pages, the planner will not put both beginning material and ending material on the same page. In fact, we put a partial order on the set of *CO*s to represent learning dependencies. $CO_1 < CO_2$ denotes the fact that CO_1 has to be learned before CO_2, because understanding CO_1 is a prerequisite for understanding CO_2.

For example, to understand the *CO "installation with televantage"*, it is necessary to know about the *CO*s *"adding user to a domain"* and *"configuring the MS outlook"*, thus:

Configure_outlook < adding_user_domain < install_with_televantage

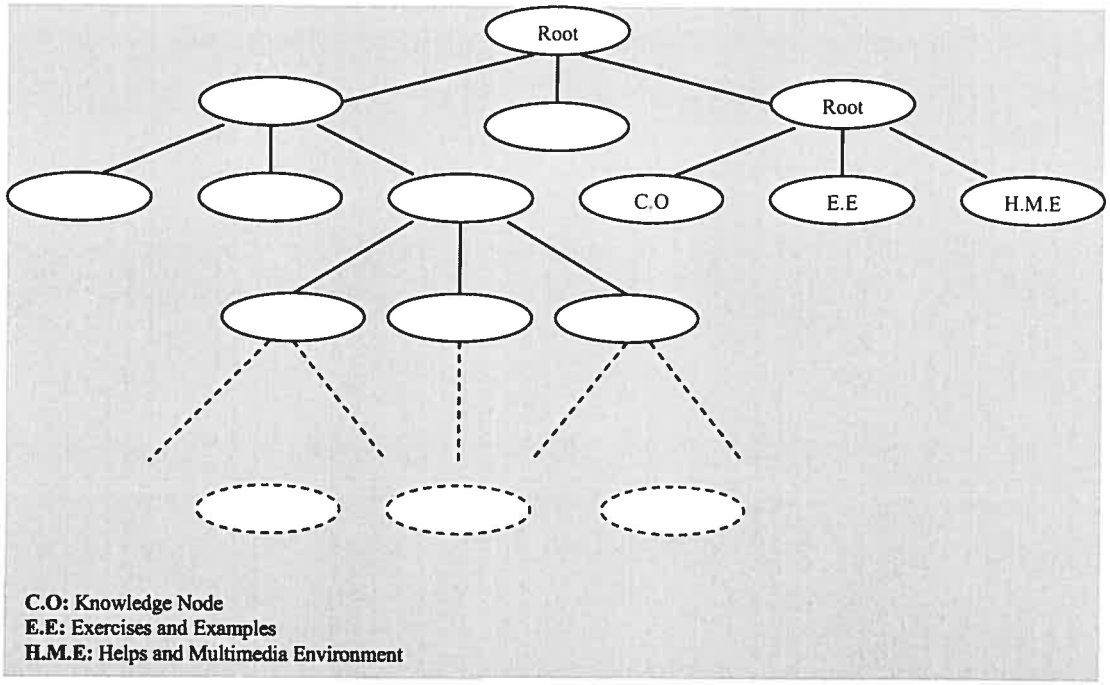


Figure 5.6 Tree of the knowledge Nodes

5.5.3 Concepts

The basic content objects that are part of the beginning and ending material for a topic are always given to each student. We call these anchor content objects. However, these objects can be augmented, thus allowing for some dynamic content generation.

This augmentation comes in the form of concepts. A concept is a collection of related content objects. Some example concepts in the Excendia are, *FrontDesk, TeleCalendar, and HelpDesk.* The idea of a concept is used since the same word or phrase may appear at many places in a course. However, these words and phrases may not have entire topics dedicated to their explanations, and some students may not know what those words mean. Therefore, we wanted a way of defining these words and phrases so that at the different times they appear, the planner can decide whether to teach that phrase, and if so, how well to present the information. When a given content object is being evaluated for use, some decisions about that object need to be made. The content object itself may have some stretch-text or some additional information. Our domain construction allows for this recursive adaptation of content objects. This recursive adaptation, however, may change the "level of difficulty" of the object, either making it harder by including more complicated concepts, or making it easier by explaining some concepts. A content object that in its base form may be too hard may get easier after it is adaptively planned. However, we are currently not considering the change in this possible effect on the level of difficulty. By using concepts, we can do more sophisticated adaptive content than simply hiding or showing certain pieces of information.

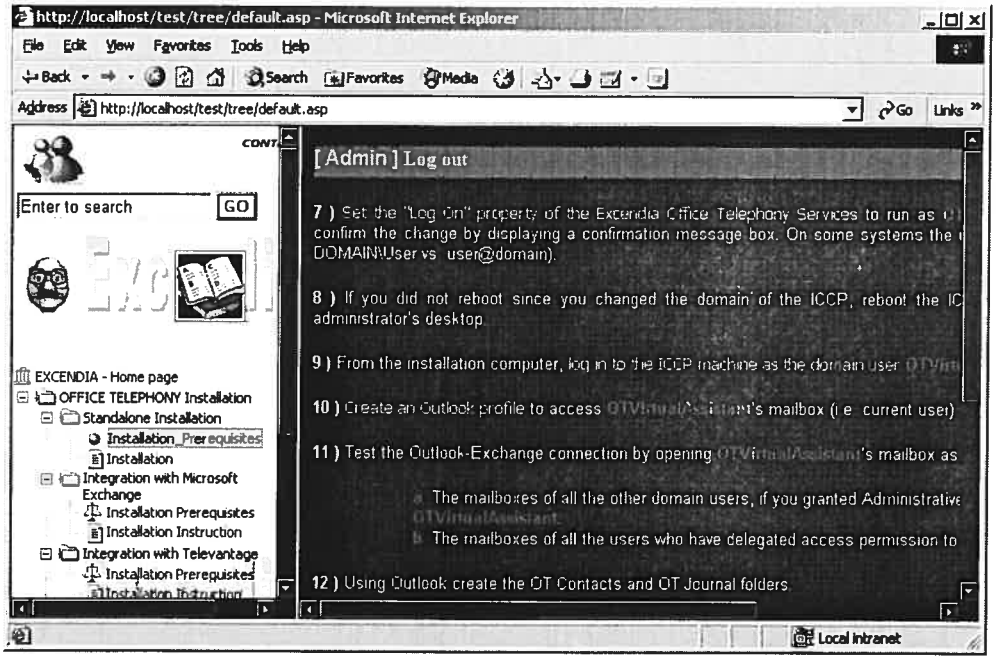


Figure 5.7 A sample page of BAHM

5.5.4 Test questions

Tests are provided as a way for students to self-regulate their learning, as well to provide the planner with the most accurate reflection of student knowledge. Tests are given when a topic is completed or whenever a student decides to take one on her own accord.

BAHM courses can have two types of questions: doing the task inside the virtual environment and multiple choice questions. In fact, student will be asked to do the action to pass the test.

Associated with each question are concepts and/or topics on which the question is testing as well as a "level of difficulty". The instructor determines the level of difficulty for the questions at authoring time.

In Figure 5.8, you may see a test sample of level of difficulty 0. User is asked to create a mailbox for the MS Exchange server.

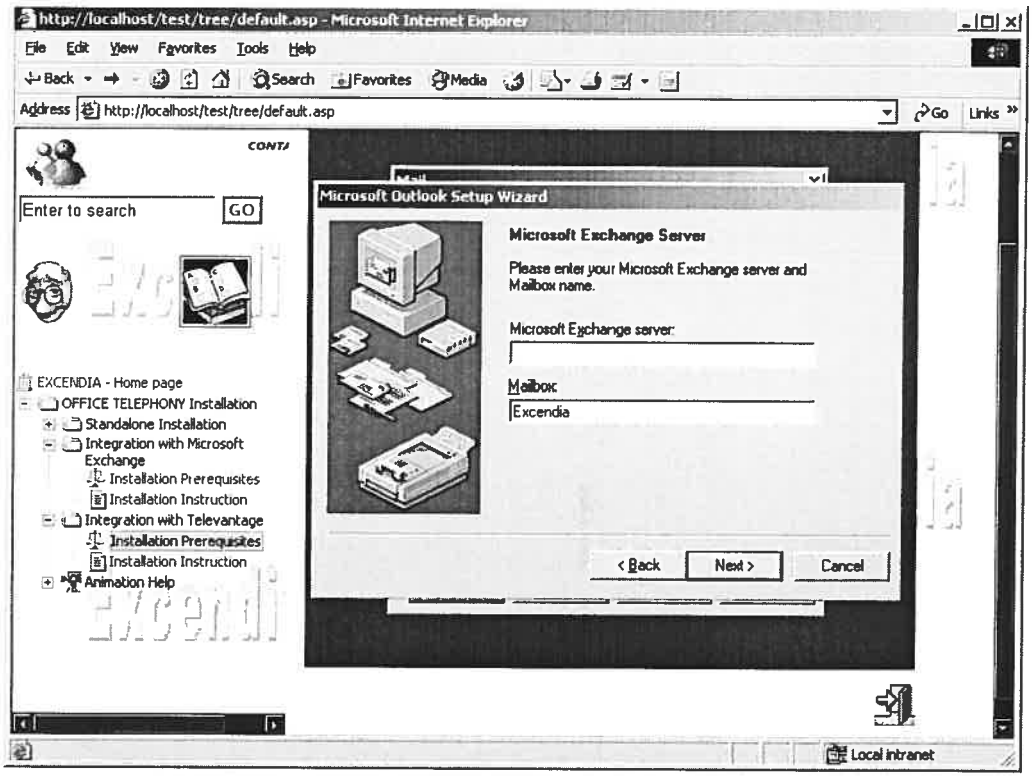


Figure 5.8 test sample of the BAHM system

5.6 Modeling the student in BAHM

In this part, we discuss the student model in BAHM, including what is represented in the model, how the model is used to personalize the interactions with the student, and how the model is updated based on student actions.

5.6.1 What is represented in the student model

The planner records information about the student's knowledge on both, topics and concepts, as well as the student's preferences for showing supplemental objects. The model of student's knowledge is a variation of an overlay model [Carr, Goldstein 1977]; recording which topics and concepts the student knows and does not know. The model of preferences records items the student desires to see and those that she does not.

In BAHM the knowledge of a user is modeled as a knowledge vector (KV). Each component of the vector is conditional, describing that a user U has knowledge about a topic KN, on the base of all observations *E* the system has about *U*:

For example, user Alex has knowledge about MS Exchange, ICCP and these facts will be considered by the system as a vector.

Definition (Knowledge Vector) KV:

$$KV(U) = (P(KN_1 \mid \varepsilon), P(KN_2 \mid \varepsilon) \ldots P(KN_n \mid \varepsilon))$$

Where $KN_1 \ldots KN_n$ are the knowledge items of the application domain and ε denotes the evidence the system monitors about U's work with the system. Observations about the student's work with the system are stored in database.

5.6.1.1 Topics

In order to suggest topics for students or to pick a topic for the student to study, the planner must track and record her performance on the topics she studies.

While interacting with a BAHM course, students gain knowledge about topics by reading the topic material, watching the video animation of the action to be learned, or doing the action related to that specific topic in the virtual environment. The students demonstrate their knowledge by answering the tests on those topics. Therefore, we need to record both how students study and how they perform on the test. We then need to synthesize this information into a judgment of the student's knowledge of those topics.
To do this, the planner records four scores on each topic.

- Studied: records how much time student has spent reading, or doing the topic material, or action related to that topic.
- Tested: records the test performance – which questions were correct and which ones were incorrect.
- Reviewed: considers if the topic has been studied more than once.
- Learned: a combination of the other three. It records how well the topic is learned. This learned score is used to determine if a student need to review a topic, and it is also provided to the student in the tree view which is available on the left side of screen so that she can track her progress (left panel in figure 5.8).

The planner computes one other score on topics, the ready score. This score indicates how ready a student is to study each topic. It is computed based on the learned scores of all the pre-topics of the present topic.

5.6.1.2 Concepts

Students interact with concepts much in the same way as they interact with topics. They read the information or watch the animation or do the action related to each concept and then answer the tests about the concept. We therefore, need a method to judge the student's knowledge on a concept.

In order to do this, the planner records a mastered value for the concept on each level of difficulty in the domain. With this, the planner can reason about the probability that the student has mastered the concept at a specific level of difficulty. These individual probabilities give the planner more fine grained information that allows it to make more

informed decisions. We discuss about how the planner uses these level of difficulty probabilities to select the most appropriate content for the user in section 5.6.2.2.1.

5.6.1.3 Student performances

Since one of the bases for adapting the presentation is the student's preferences, we must store these in the student model. To do this, we record the features which student prefer by asking her the questions and giving her the options to choose. We also record information about the current topic, the current concept, the tests the student has passed, how many times the concept has been presented, etc. this gives us a snapshot of the student's learning at the time she decides to see or hide an object. The BAHM also may not hide the object, but it makes the object dynamic – which makes the object adaptive to individual user. These additional properties may be useful in learning patterns about the user's behavior.

Student performance observation is expressed with four grades – the grades of knowledge acquired by the student.
A student can have

- "expert's knowledge" on a CO rating E: excellent,
- "advanced knowledge" rating A: with some difficulties but mainly excellent,
- "beginner's knowledge" rating B: with many difficulties - seems not to master the concept, or topic,
- "Novice's knowledge" rating N: not ready for this concept/topic yet.

which we have assigned a value each. Thus, the COs are, on the one hand, concepts describing the application domain of the topics, on the other hand, they are random variables with the four discrete values E, A, B and N, coding knowledge grades. The evidence we obtain about the student's work with the system changes in time. Normally, the student's knowledge increases while working with the system, although lack of knowledge is equally taken as evidence. Since every kind of observation about a student is collected as evidence, the knowledge vector gives – at each moment – a snapshot of the student's current knowledge.

5.6.1.4 Other sorts of knowledge in the student model

We do not agree with John Self who claims that the student model should not collect more data than the tutoring component can use [Self, 1990]. We take the opposite approach: the student model in BAHM collects everything it can, even if it is not necessarily used by instructional component.

Our reason for collecting a lot of information is that running studies with human users is extremely time consuming and expensive and it can be very difficult to find a group of suitable users. In addition, at the time when we run the user study, we may not have planned all of the reasoning about the apprentice the instructional component can do. Therefore, while we have a group of users using the system, we will gather as mush information about them as we can. We can then use this actual user data to improve the instructional component and cause it to be adaptive to even more student behavior characteristics than currently possible. We also keep a record of each test taken and each answer given (score acquired). This information is actually used in the BAHM, as questions that have been answered correctly are not given to the student again.

5.6.2 Using the student model

In this section, we discuss how the student model is used to make decisions in the BAHM system. The model is used for four main tasks: adaptive navigation (i.e. curriculum sequencing), adaptive content presentation, adaptive testing, and creating the dynamic links.

5.6.2.1 Adaptive navigation

In adaptive hypermedia systems with non-linear paths through the domain, students run into the "lost in hyperspace" problem. Students are not sure which paths they should follow, and when they follow a path, they are not sure how to find their way back to the beginning. Adaptive navigation techniques help solves these problems.

In BAHM courses, students might need help, determining a good sequence in which to study topics. In this section, we discuss the solutions to this problem in the BAHM system. The student can either let the planner choose the next topic to study or she can choose it herself. If the latter option is chosen, the planner provides the student with information advice on good paths through the curriculum.

5.6.2.1.1 Rules for moving to a new activity

When the student reaches the end of a topic, and she uses the "Next" button, the planner must determine the next material for the student to see. This material may either be a survey, a test, or a new topic.

In order to ensure that the student answers our surveys, the planner first checks if it is the time to give a survey (half of the topics have been seen or the course is complete), and if so, the survey is given. If it is not time for a survey, but the student has seen too many topics without taking a test, then a test is presented. We have chosen to give a test at this time so that the student does not see too many topics before a test is resented, therefore be more susceptible to information overload.

If neither of these conditions is true, then the planner must decide on the next student interaction. The algorithm to pick the next topic is designed to keep the student's momentum through the curriculum. The planner first looks for a topic to review, and then looks for a new topic that is ready to teach. It will not choose a topic that was taught since the last test, since testing gives the most evidence of mastery and a topic should be re-taught until a test has been given.

Topics designed for repetition have a higher priority over new topics to study. The reason for this is that students should understand basic material before going on to later topics. Also, at some point, these topics will need to be reviewed, and we feel they should be reviewed as soon as possible.

To find a topic that needs review, the planner performs a breath first search from the last topic taught, searching both forward and backward from that topic. A topic needs review if 1) it has been studied once before and 2) it has not been mastered. If such a topic is found through this search, it is taught to the student. However, if a topic is chosen for repetition, it is possible that a remedial topic will be taught first before repeating the topic.

In this case, simply reading the same text, seeing the same graphics or doing the same action in the virtual environment again will most likely not result in success.

If no topics need review, the planner uses a similar breath first search from the last topic taught to search for a topic that is ready to teach. If such a topic is found, that topic's root ancestor is taught. The reason to teach the ancestor along with the topic itself is to give the chosen topic context in the topic structure.

If no topic is found through either of these searches, the planner will go up to the parent of the last topic taught and attempt the searches again. Still if no topic is found and a test has not just been given, the planner will give a test. This condition arises if a topic is taught, but is not mastered through studying, no topic itself will need review and no topics are enabled. After the test is given, either the topic itself will need review, or another topic may be found that can be taught. If a test has just been given and no topic in the graph can be found, then any topic that needs review is taught [Cini, Valdeni,de Lima, 2002]. If there are no topics to review, a ready topic is found.

If the student decides to choose the next topic, the planner presents the topic net to her, annotated with suggestions from the planner. The student is informed of which topics have been mastered, which topics should be repeated, and which topics she is ready to start. For each of these topics, the student is given both the "ready" score and the "learned" scores which the planner has calculated, as a guide for choosing her next topic (table 5.2).

This is similar to the adaptive annotation technique used in ELM-ART [Brusilovsky, Schwarz, Weber, 1996].

Table 5-2 Sample of Student Ranking

Node	Link type	Threshold Needed	Student 1	Student 2	Student 3
A	Prerequisite	0,8	0,75	0,81	0,30
B	Prerequisite	0,8	0,99	0,85	0,50
C	Related	0,5	0,99	0,51	0,90

5.6.2.1.2 Computing "ready" scores

In the previous section, we discussed in general when the planner needs to pick a new topic for the student to teach. In this section, we give the specific algorithm the planner uses. Essentially, the planner must determine which topics are "ready" to be taught. A topic's "ready" score is based on the "learned" scores of both its pre-topics' and its ancestors' pre-topics. The "learned" scores on the pre-topics are important when considering what new topics to present. In order to start a new topic, a student should show sufficient knowledge of its prerequisite material. Most systems require mastery of all prerequisite material before the student can proceed to new topics. Another way of stating this criterion is that the mastery values on all prerequisite topics must be above a *certain threshold*. However, in BAHM, we do not necessarily impose a strict threshold on the grades needed on each related topic. Consider a topic which is connected to three pre-topics (A, B and C) and three different students' topic ratings on these three pre-topics (Table 5.2).

If strict threshold such as those shown in table 5.2 are used, then student 1 would not be qualified to start the new topic, but student 2 would be. However, it is not clear that student 2 is any more qualified than student 1 to start the next topic. Therefore, we decided not to use strict threshold, but rather to use a more flexible method for determining a "ready" rating for a topic.

Each topic's "learned" score is used to determine the weight on the link between that topic and others in the semantic network. The link weights take into account the link type because for each different link type, the level of mastery needed to move on in the curriculum may not be the same. For instance, a student should demonstrate more knowledge on a strict prerequisite than on just a related topic. Therefore, the thresholds required on different link types are different.

We have constructed rules that adjust the link weights to take into consideration the link types, as shown in table 5.3. The weights are adjusted based on how close the "learned" score is to the threshold. If the pre-topic's "learned" score is above the threshold, the adjusted link value is 1. After the conversions, the weights range from 0 to 1, inclusive. By using these kinds of scaling rules, we give more weight to different kinds of relationships. The remedial topic rules are used only if the remedial topics have in fact been taught.

Table 5-3 Link update rules

Link type	Threshold	Learned value within x of threshold	Updated link value
Prerequisite	0,8	-0,10	0,8
		-0,30	0,6
		-0,50	0,4
		-0,70	0,2
Related	0,5	-0,10	0,7
		-0,25	0,4
		-0,35	0,2
Co-requisite	0,65	-0,10	0,8
		-0,30	0,5
		-0,50	0,2
Remedial	0,8	-0,10	0,8
		-0,35	0,5
		-0,60	0,2

Once the weights of links are determined, we must then compute the ranking on the topic. This can simply be done by taking an average of the adjusted link weights of the pre-topics of the topic in question.

For example, consider a topic that is linked to topics A, B and C, and consider the values for three students on these topics (Table 5.2). Student 1 is very close to the threshold for topic A and over the threshold for topics B and C. Student 2 is over the threshold for all three topics, and student 3 is under the threshold for topics A and B and over it for topic C. As we would expect, using the rules in Table 5.3, student 2 has the highest ranking ($1 = {}^{(1,0+1,0+1,0)}/_{3}$), followed by student 1 ($0,933 = {}^{(0,8+1,0+1,0)}/_{3}$), and finally student 3 ($0,667 = {}^{(0,4+0,6+1,0)}/_{3}$).

5.6.2.2 Adaptive content presentation

As we have said, the main interaction that students have with BAHM courses is by doing the appropriate action inside the virtual environment which, is presented to the student adaptively. However, each presentation (Virtual action to be done) must be created

dynamically since not all students will need or want to see the same material. One popular technique used for the adaptation of material is to have a set page with places indicated where material or dynamic environment can either be added or omitted, either through variants [Fink, Kobsa, Nill, 1998] or through stretchtext [Boyle, Encarnacion, 1998]. However, using this technique can lead to either very short pages or very long ones. The former is not appropriate since it can lead to frustration at having to turn the page frequently and lack of cohesiveness of the presentation. The latter is not appropriate since it can lead to information overload and from the point view of design, users should not be forced to scroll down or up to see all the material inside a web page.

Therefore, BAHM dynamically determines how much extra material will be shown. Furthermore, we want to create the pages which contain the interactive multimedia environments dynamically as they are requested, i.e. we do not want to design all the pages for an entire topic at the beginning of the topic. We want the actions a student performs early in a topic presentation to affect the creation of the pages that will be shown later. For example, if a student on the first page of a topic asks to see a lot of material that would normally come later, we want to avoid showing the same material at that later time.

5.6.2.2.1 First phase – student's knowledge

The strategy of the planner is to give supplemental material that is neither too hard nor too easy for the student. Therefore, the planner must have a method for evaluating content objects for this purpose.

Supplemental content objects have a "Level of difficulty" ranging from 0 to 3 (corresponding to easiest, easy, medium, and hard), which describes how hard they are to understand. This level of difficulty measurement is important to enable the content to be adapted based on the student's knowledge.

When deciding what supplemental content objects are at the correct level of difficulty, the planner must reason about the student's knowledge of the concept. Each concept has four different mastery values, one for each level of difficulty. These mastery values are determined by a *pre-test*. Thus students with different abilities start the course with different *a priori* mastery values.

To decide which content objects are at the correct level of difficulty, the planner simply determines the highest level of difficulty the student has mastered. A level of difficulty is said to be mastered if its mastery value is greater than 0,85 (on a 0 to 1 scale). The planner then chooses those content objects that are one level of difficulty higher than the student's highest mastered level of difficulty. Of course, if the student has not mastered level 0, then level 0 is chosen as the correct level of difficulty. Also, if level 3 is mastered, then no additional objects need to be shown, since the concept is mastered.

5.6.2.2.2 Second phase – student's preferences

The second phase of the decision making involves selecting content objects that the student will *want*. As we talked about Kolb's theory, one of our goals is to be able to generate a presentation that is satisfactory to the student in terms of her learning styles and preferences. However, in order to do this, the system needs to learn what those learning styles and preferences are.

Each content object has a set of features (see table 5.4). The planner classifies each object that could be presented to the student based on these features. The first three features in table 5.4 describe basic characteristics of objects as defined by the author. The fourth attribute, PlaceInTopic, identifies if the object is in the BeforeContentObjects or the AfterContentObjects of the topic. We have included this attribute since students have different learning styles, and may want certain kinds of objects at the beginning of a topic presented differently than those they would want at the end of a topic. For example, some students like abstract explanations given first, followed by concrete video animation example, while others like examples followed by explanations. This PlaceInTopic attribute can help the planner better design the order of presentation within a topic. The fifth attribute is similar to PlaceInTopic, but it considers how many times the concept has been presented with respect to the total number of times it could be presented in an entire course. This attribute also helps determine if the user has presentation ordering preferences.

Table 5-4 Features in BAHM system

Features	Values
Media types	{Graphics, Video animation, Interactive multimedia}
Instructional type	{Explanation, Example, Description, Definition}
Abstractness	{Abstract, Concrete}
Place in topic	{Beginning, End}
Place in concept	{Beginning, Middle, End}
Wanted	{Yes, No}

The last attribute, *wanted*, describes whether the object itself was wanted by the student. This can be determined by observing the student's reaction to having the object presented or not presented. If the object was presented and was not hidden by the student, or if the object was not initially presented but was requested, then the object is considered *wanted*. An object is not *wanted* if it was presented to the student and hidden, or if the link to the object was not used.

The planner "learns" a student's preferences by observing student actions and applying machine learning to the task of interpreting those actions. A Naïve Bayes Classifier is relevant to problems where each instance x can be described by a combination of attribute values and where the target function $f(x)$ can take on any value from a finite set V. When a new occurrence is obtainable a value for the target function can be calculated based on the training instance. The machine leaner choose the value for the target function that has the highest probability, based on the training set.

The attribute of the Naïve Bayes Classifier is as $<\alpha_1 \ldots \alpha_n>$:

$$\upsilon_{NB} = \underset{\upsilon_j \in V}{\operatorname{argmax}} P(\upsilon_j) \prod_i P(\alpha_i|\upsilon_j) \qquad (5.1)$$

(See the example on page 82).

A Naïve Bayes Classifier is the chosen learning method because it is fast (i.e. can work well in real time) and has been proven to be very accurate for user modeling tasks

[Billsus, Pazzani, 1997]. It also works well with little data, which is a consideration with a user modeling application. We want the machine learner to be able to start learning about the student after very few interactions. Since the goal of the adaptive content is to provide a presentation that does not require any changes by the student, the Naïve Bayes Classifier is used to predict if an object will be wanted or not, based on the other features of that object. Those objects that are predicted to be wanted will be shown to the user, while the others will not be shown. In order to determine which content objects to show, the planner first groups all content objects at the correct level of difficulty, as described in the last section, with the same features into a feature class. If these objects are shown, they will be shown as a group, in order to simplify the interface for the student. Using this grouping technique, the student does not have to ask, for example, to see all abstract picture examples at level of difficulty to find the one he wants. After this grouping had been done, the planner examines each content object in question, and uses the Naïve Bayes Classifier to predict if the object will be *wanted*. The Naïve Bayes Classifier works by essentially comparing the features of the current content object in question to features of content objects in its instance space (see table 5.5). The instance space consists of content objects that, in the past, were either wanted or not wanted by the student.

In BAHM, we use the Naïve Bayes Classifier as follows. If we have an object that has features:

< MediaType = video animation, Instruction Type = Definition, Abstractness = Concrete, PlaceInTopic = begining, PlaceInConcept = Middle>

Then we would consider which one of the following two formulas is greater:

$$\begin{aligned} &P(Wanted = yes)\ x\ P(MediaType = Video\ Animation|\ Wanted = yes)\ x \\ &\quad P(InstructionType = Definition|Wanted = yes)\ x \\ &\quad P(Abstractness = Concrete\ |Wanted = yes)\ x \\ &\quad P(PlaceInTopic = Before\ |Wanted = yes)\ x \\ &\quad P(PlaceInConcept = Middle\ |Wanted = yes) \end{aligned} \qquad (5.2)$$

$$\begin{aligned} &P(Wanted = no)\ x\ P(MediaType = Video\ Animation|\ Wanted = no)\ x \\ &\quad P(InstructionType = Definition|Wanted = no)\ x \\ &\quad P(Abstractness = Concrete\ |Wanted = no)\ x \end{aligned} \qquad (5.3)$$

$$P(PlaceInTopic = Before \mid Wanted = no)\ x$$
$$P(PlaceInConcept = Middle \mid Wanted = no)$$

Each of these individual probabilities can be calculated very easily. The probability that an object is wanted is the ratio of the objects that were wanted to all objects in the instance space.

In order to calculate the conditional probabilities, we must break the instances into two sets: those with *wanted = yes* and those with *wanted = no*. For equation 5.2, we count how many items in the set wanted = yes also have the attribute value in question. For example, P (MT *Video Animation* |W = y) is the number of times that of all wanted objects; the object was a Video Animation object. For equation 5.3 we count how many items in the set wanted = no have the attribute value in question. Thus, P (MT = *Video Animation* |W = n) is the number of times that objects that were not wanted were Video Animation objects. This count must be done for each of the attribute values in the candidate object. By using these measures, we can judge how important each attribute is to the student, and also how the combination of features represents the student's preferences. The planner can then choose objects whose attribute set is most preferable to the student.

Even if the user model is turned off, these predictions are made and recorded. In this case, the classifier is asked to predict which objects will be wanted or not wanted. However, when there is no user model, these predictions are not used to alter the presentation. These predictions are only recorded so we can determine how accurate the classifier is. If the classifier predicts an object would be wanted and the student elects to show that object, this is considered an accurate prediction. Thus even without a user model, when student explicitly changes the presentation, we can judge the accuracy of the classifier.

Practical Example:

Let's consider a student whose instance spaces is as seen in Table 5.5

Table 5-5 A sample instance space

Instructional Type	Media Type	Place in Topic	Place in Concept	Abstractness	Wanted
Example	Text	Beginning	Beginning	Abstract	Yes
Definition	Text	Beginning	End	Concrete	Yes
Definition	Text	End	End	Concrete	Yes
Description	Text	Beginning	Beginning	Concrete	Yes
Explanation	Animation	Beginning	Middle	Abstract	Yes
Definition	Text	End	Middle	Abstract	No
Description	Animation	End	Middle	Abstract	No
Explanation	Animation	End	End	Abstract	No
Example	Animation	Beginning	Beginning	Concrete	No

The student also has the following mastery levels on the concept *switch*:
level 0 = 0,62, level 1 = 0,46, level 2 = 0,33, level 3 = 0,1

While planning the next page, the system comes across the text object that consists of the phrase "a set of computers and/or switches connected by communication links." In this phrase, the word "switch" refers to the concept *switch*. Therefore, the system must decide which content objects that are associated with that concept to include for the student. The possible content object features are given in Table 5.6.

Table 5-6 Content objects in concept *switch*

Name	Instruction Type	Media Type	Place in topic	Place in concept	Abstractness	Level of Difficulty
Switch 1	Explanation	Animation	Beginning	Beginning	Concrete	0
Switch 2	Description	Animation	Beginning	Beginning	Abstract	0
Switch 3	Explanation	Animation	Beginning	Beginning	Abstract	1

The first phase of the algorithm determines which content objects are at the correct level of difficulty. Based on the algorithm given in section 5.6.2.2.1, the correct level of difficulty is 0. Therefore, switch 3 can be eliminated, but both switch 1 and switch 2 are at

the correct level of difficulty. The system then uses the Naïve Bayes Classifier to determine if either of those objects is *wanted.*

Using the Naïve Bayes Classifier, the system must determine if switch 1 will be wanted by the student. Equation 5.4 (based on equation 5.2) is used for this purpose to determine P(Switch 1 Wanted = yes)

$$P(Wanted = yes)\ x\ P(MediaType = Animation \mid Wanted = yes)\ x\ P(InstructionType = Explanation \mid Wanted = yes)\ x\ P(Abstractness = Concrete \mid Wanted = yes)\ x\ P(PlaceInTopic = Beginning \mid Wanted = yes)\ x\ P(PlaceInConcept = Beginning \mid Wanted = yes) \quad (5.4)$$

Which numerically is:

$$5/9 \times 1/5 \times 1/5 \times 3/5 \times 4/5 \times 2/5 = 0{,}004266 \quad (5.5)$$

Equation 5.6 (based on equation 5.3) is used to determine P(Switch 1 Wanted = no)

$$P(Wanted = no)\ x\ P(MediaType = Animation \mid Wanted = no)\ x\ P(InstructionType = Explanation \mid Wanted = no)\ x\ P(Abstractness = Concrete \mid Wanted = no)\ x\ P(PlaceInTopic = Beginning \mid Wanted = no)\ x\ P(PlaceInConcept = Beginning \mid Wanted = no) \quad (5.6)$$

Which numerically is:

$$4/9 \times 3/4 \times 1/4 \times 1/4 \times 1/4 \times 1/4 = 0{,}001302 \quad (5.7)$$

Normalizing these values, we get P (*Switch 1 Wanted = yes*) = *0,766* and P (*Switch 1 Wanted = no*) = *0,234*. Therefore, Switch 1 would be shown. With similar calculation, we will get the result for Switch 2 as follows:

The probability that Switch 2 is wanted is calculated using equation 5.8.

$$P(Wanted = yes)\ x\ P(InstructionType = Description \mid Wanted = yes)\ x\ P(MediaType = Animation \mid Wanted = yes)\ x\ P(PlaceInTopic = Beginning \mid Wanted = yes)\ x\ P(PlaceInConcept = Beginning \mid Wanted = yes)\ x\ P(Abstractness = Abstract \mid Wanted = yes) \quad (5.8)$$

Which numerically is:

$$5/9 \times 1/5 \times 1/5 \times 4/5 \times 2/5 \times 2/5 = 0{,}002844 \quad (5.9)$$

The probability that Switch 2 is not wanted is calculated using equation 5.10.

$$P(Wanted = no)\ x\ P(InstructionType = Description \mid Wanted = no)\ x\ P(MediaType = Animation \mid Wanted = no)x\ P(PlaceInTopic = Beginning \mid Wanted = no)\ x\ P(PlaceInConcept = Beginning \mid Wanted = no)\ x\ P(Abstractness = Abstract \mid Wanted = no) \quad (5.10)$$

Which numerically is:

$$4/9 \times 1/4 \times 3/4 \times 1/4 \times 1/4 \times 3/4 = 0{,}0039 \quad (5.11)$$

P (*Switch 2 Wanted = yes*) = *0,421* and P (*Switch 2 Wanted = no*) = *0,579.* Therefore, Switch 2 would not be shown.

5.6.2.3 The tests

In this section, we discuss how the tests are selected. We also discuss the type of feedback the student receives after a test is given and graded.

As we have mentioned, tests are already generated with Flash MX, the planner decides which test should be asked. Each test in the database is evaluated to see if it is appropriate for the user at the specific time. The factors the planner takes into

consideration are if the test is at the correct level of difficulty and if the test covers the *correct* material. The "correct" material may be the tests covering the topics and concepts that are not mastered as well as topics and concepts that should be mastered and should be reviewed.

For choosing the test related to concepts, the planner must determine the highest mastered level of difficulty. We use the technique given in the section 5.6.2.2.1 to determine the student's highest mastered level on a concept. The planner then presents the test that is one level higher than the highest mastered level. Two boundary conditions must be considered. The first is for students who have not mastered any levels. They are given level 0 test.

The second is for students who have mastered all levels of the concept. Those students are not given any tests concerning this particular concept. However, for the post-test, we need a different technique. If we restrict the student to answering only questions on a single level of difficulty, she has no chance of mastering the topic based on those tests alone. This is the case since our test score updating technique requires many tests to be done correctly before mastery is reached. Unless there are an extremely large number of questions at a given level of difficulty, which is usually not the case, she will not be able to reach mastery without seeing tests at other levels of difficulty. As a result, the student would be forced to review the topic, even if she knows all the material. Thus we need a different method for testing students on topics.

The method we have chosen is to present the tests that are at levels of difficulty greater than the student's mastery level on the topic. This requires a way for the planner to map a student's mastery value to a level of difficulty. The mapping is given in table 5.7.

Table 5-7 Mapping between mastered value and level of difficulty

Mastered value	Level of difficulty
Greater than 0,8	4
$0,6 \leq X < 0,8$	3
$0,4 \leq X < 0,6$	2
$0,2 \leq X < 0,4$	1
X less than 0,2	0

As a general rule, when picking tests on concepts and post-tests that are not mastered, the planner does not give tests that have already been answered correctly. The planner may elect to do some review work on topics on which the student has done well. If the test score on a topic is above a certain threshold, the planner then looks at how long it has been since the topic had been part of the test. If the time has been sufficiently long, then the topic is tested. The time variable is a function of the topic's score: the higher the score, the longer time interval allowed. The reason for this is that if the score on the topic is very high, there is a good chance the student knows the material and will thus need less frequent reminders. When testing on a topic that has a high score, the planner tries to choose the test that is harder than those asked in the past. If none exist, those tests that have not been asked for the longest period of time are chosen.

After a student takes a test, she is told which part she answered incorrectly. However, she is not told the correct answer. Rather, she is instructed to recover the associated material. The incorrect tests will then be presented to the student in future tests.

5.6.3 Updating the student model

In this section, we discuss how the student model is updated based on the student's interactions with the planner. We discuss the grading of concepts and topics as well as the recording of student's preferences.

5.6.3.1 Timing information

We are considering timing information for grading both concepts and topics. Therefore, we need a way to judge how much time the student has spent on a page, and even more fine grained, how much time she has spent on each content object on the page. As a result, when each page is shown, we need a method to allocate the time that the page was visible to the components of that page.

Each content object takes up a certain number of lines on the screen. We can therefore allocate a "*time per line*" by diving the total time the page is showing by the total number of lines on the screen, calculated by summing the lines of all the content objects. Each object can then be allocated the time per line multiplied by the number of lines of the object. However, this technique only works for the static pages, i.e. those whose content does not change. Consider the case when an object is asked to be shown 10 second after the page is initially shown. Now how should the time be allocated?

To solve this problem, we consider the time between page events, where events including showing or hiding an object and leaving the page. When an event occurs, we divide the time since the last event evenly amongst the lines on the screen. For objects that remain on the screen when events occur, we sum the time per line allocation over those multiple events.

For example, say object A is 2 lines long and is initially the only object shown on a page. After 10 second, object B is shown, which is 5 lines long. At this point, object A gets a time value of $(^{10\ Sec}/_{2lines})$x 2 *lines*. Now after another 10 seconds, object B is hidden. At this point, object A gets $(^{10\ Sec}\ /\ _{7lines})$ x *2 lines* amount of time added to its total, while B gets $(^{10\ Sec}\ /\ _{7lines})$ x *5 lines* amount of time. After another 10 seconds, the student moves to the next page, and A gets another $(^{10\ Sec}\ /\ _{2lines})$ x *2 lines* worth of time. In sum, A has 22,857 seconds worth of time and B has 7,143 seconds worth of time.

5.6.3.2 Concepts

As we have indicated, each concept has a score for each level of difficulty. This score reflects the probability that the student has mastered that level of difficulty. After the student sees a content object that belongs to a concept or answers a test concerning a concept, that concept's mastered values must be updated. We discuss the updates after a content object is seen and then we discuss the updates after a test is given. With both updating, we use regression equations, a technique similar to that used by Shute [Shute 1995] for updating a concept's grade.

5.6.3.3 Content objects

We only update a student's grades for content objects that are seen. The reason for this is we do not want to penalize students for not seeing the objects the planner predicts that they will not want to see. In addition, the student does not gain any knowledge from not seeing material. The concept grading occurs when either (1) as soon as an object that was seen is hidden or (2) when the page changes. In the latter case, we observe which content objects are being viewed on the page just left, and update their related concepts. The page can be either a courseware content page (as the student is reading) or a glossary page.

Every mastery level is updated for each content object seen. For example, say a student sees a text object of level of difficulty 2. This affects the probability that each level of difficulty is mastered. Specially, there is considerably more evidence that the student has mastered levels of difficulty 0 and 1, some evidence that level of difficulty 2 is mastered, but little evidence that level of difficulty 3 is mastered. Thus, all four levels of difficulty values are updated.

We use regression equations to determine how the grades should be updated. Each regression equation determines a new value for the probability that the level is mastered, based on the old value. Furthermore, each equation is responsible for calculating new values for a single level of difficulty and for a single event. For example, to update the student's level 0 mastery, we need equations that represent if a level 0 object was seen, a level 1 object was seen, etc. Thus for each level to update, we need 4 equations (there are four levels of difficulty).

There are 4 equations to determine the level 0 mastery, 4 equations to determine the level 1 mastery, and so on, for a total of 16 equations. However, this total of 16 equations does not take into consideration how much time the student has spent studying a content object. For simplicity, we use a binning method to record the amount of time spent, as using the real number values would prove too cumbersome for regression equations. We have decided on three bins: too little, average, and too much. Therefore, we actually use 48 equations, 16 for too little time, 16 for average time, and 16 for too much time. The time value used is simply how much time the user viewed the object this time; it is not a total of all time ever spent viewing the object. These regression graphs can be found in appendix A.

The general idea for updating concept grades based upon which content objects are viewed is as follows:

- To update the mastery score for level of difficulty x when seeing an object of level of difficulty ≥ x, increase the score more when the optimal time is spent on the object and increase the score less when the object is seen for too little or too much time.
- To update the level of difficulty x mastery score when seeing an object of level of difficulty < x, decrease the score for an object seen for not enough time, more for an object seen for optimal time, and even more for an object seen for too much time.

The reason for the first rule is that if the student sees something harder than what she has mastered, she should be rewarded for that. The reasoning for the second rule is that if the student sees something easier than what she has mastered, the planner has misjudged her knowledge. However, if she only briefly looks at the object, she should not be penalized the same as if she had to study the object for a longer time.

5.6.3.4 Tests

Just as with content objects, we can adjust a student's mastered probabilities after the tests. Both questions the student has answered correctly and the ones she has answered incorrectly are used to update the values.

However, unlike with studying content objects, each level of difficulty does not necessarily need to be updated. For example, if the student answers correctly a question of a level of difficulty 0, this gives no evidence about her knowledge of material of level of difficulty 3. However, if she gets a question correct at level of difficulty 2, that does imply something about her level 0 knowledge. Furthermore, the larger the difference between the level of the correct question and the mastery level to be updated, the more that mastery is increased. The reason for this rule is that getting a level 3 question correct is much harder than getting a level 0 question correct, and thus provides more evidence that level 0 is learned.

In similar vein, if a student gets a level 3 question incorrect, that does not imply anything about her knowledge below that level. However, if she gets a level 0 question incorrect, her knowledge of levels above that level should be adjusted. Just as with correct questions, the larger the difference between the incorrect question's level and the level to be updated, the larger the decrease in the mastery value. For example, a student who has mastered level 3 should not be getting level 0 question wrong. However, a student who has mastered only level 1 has a much higher probability of getting a level 0 question wrong, and thus should not be penalized as much as the level 3 student.
The general scheme for the test regression equations is:

- For a question answered correctly at level x, update the mastery values for levels less than or equal to x, with the lower the level receiving a larger increase.
- For a question answered incorrectly at level x, update the mastery values for levels greater than or equal to x, with the higher levels receiving a larger decrease.

However, these rules do not take into consideration that questions at the same level of difficulty may actually be easier or harder to answer. For example, multiple-choice questions at lod (level of difficulty) 0 should be easier to answer correctly than practical questions – questions that should be done inside the virtual environment. Thus, we need to create regression equations which consider both levels of difficulty as well as question type. So if we have only two types of questions, we would need a total of 40 equations[9].

Since we restricted our user test to a domain with only 2 levels of difficulty, we actually only needed 20 equations. These equations can be found in appendix A.

5.6.3.5 The way regression equations were obtained

We wanted to employ the help of domain experts in obtaining information on how to update a concept's grade. However, we did not feel comfortable asking experts to provide us with regression equations, as this seemed too difficult a task. Rather, we had experts

[9] 2 question types x 4 levels of difficulty to update x 4 levels of difficulty right + 2 question types x 4 levels of difficulty to update x 4 levels of difficulty wrong = 64. However, subtracting the equations that do not need to be constructed, since no evidence is given by those events (e.g. updating level 4 when getting a level 0 question correct), we arrive at 40 equations.

give us five points for the equation to be plotted. These points indicated what the new mastery value should be after an event has occurred, given the student's current mastery value. For example, for 5 different prior mastery values, we asked the expert to tell us the student's new mastery values when she has seen a level of difficulty 2 concept object for the optimal time. Once these data points were obtained, we used curve fitting techniques to determine the curve that best matched those points. We used polynomial regression [Draper, Smith 1981] to determine the best polynomial. We tested 1, 2, and 3 degree polynomial and determined the error given by those polynomials compared to the original 5 points. The curve that had the best error was used for the regression equation.

Figure 5.9 shows two different regression equations for the points (0,1 0,5), (0,3 0,67), (0,5 0,85), (0,7 0,93), and (0,9 0,99). The linear equation has an error of 0.051381 while the degree 2 equation has an error of only 0,017403.

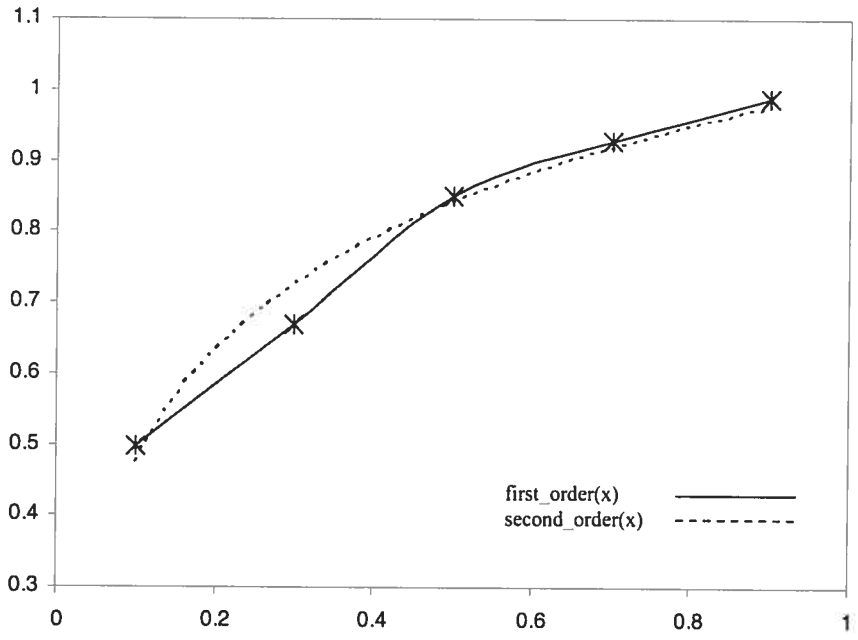


Figure 5.9 Example of two regression equations for the same points

5.6.3.6 Student's performances

In section 5.6.2.2.2, we discussed how the Naïve Bayes Classifier predicts whether an object is wanted or not. In this section, we discuss how the actual wanted values of content objects are determined.

When a student leaves a page either to see another page or to take a test, the planner analyzes which supplemental content objects were shown and which were hidden. Objects which were shown were either originally shown and not hidden or originally hidden and asked to be shown. Objects that were hidden either remain hidden the whole time or were shown and then hidden by the user.

Those objects that are shown are considered *wanted*, while those that are hidden are considered *not wanted*. The planner sets the *wanted* feature value of the objects to *Yes* if they were shown and to *No* if they were hidden. These objects are then added to the example space of the Naïve Bayes Classifier, thus increasing the size of the training set.

It should be noted that as soon as an object is asked to be shown, it is considered wanted, and as soon as it is hidden, it is considered not wanted. As a result, the same object may be considered both wanted and not wanted.

We do not penalize the student for not viewing objects that are at the "wrong" level of difficulty. Therefore, objects that are at a level of difficulty other than the one chosen by the planner that were not seen are *not* added to the feature. However, if the student chooses to see objects at a different level of difficulty, those objects are considered wanted, and are therefore added to the example space with their *wanted* feature value set to *Yes*.

It is important to note that the Naïve Bayes Classifier initially predicts the *wanted* value, but it is not that prediction that is added to the example space. Rather, it is the actual value of the student's desire that is added to the example space. The classifier will use this growing example space for future predictions.

Chapter VI

6. Grading & Evaluation

In this chapter, we discuss the method that we used for grading the students and evaluating their performance.

6.1 Grading topics

We discuss how each topic is graded, based on direct evidence. Three factors are important in determining the student's mastery of a topic:

- How well the student performs on tests on the topic
- How well it has been studied
- How much the topic has been reviewed

These three pieces of evidence are then combined to determine how well a topic has been "learned".

6.2 Test performance

In intelligent tutoring systems, tests give a tutor the most direct information about the student's knowledge. For this reason, tests are included in BAHM and are considered the most important piece of information the student model can obtain about the student.

When grading a test, the test grades of the question's associated topic and that topic's ancestors need to be updated. The ancestors are updated since a question on a subtopic is clearly relevant to the grade on the ancestors of the subtopic.

Furthermore, each question's level of difficulty is taken into consideration. Clearly answering correctly a harder question demonstrates a higher ability than answering correctly an easier question. Similarly, failing at a harder question should not be as damaging as failing at an easier question.

We are using a Bayesian updating formula to compute the student's test grade, since the grade is really a probability that the student knows the topic. The basic Bayes' formula is:

$$P(\mathrm{B}|\mathrm{A}) = \frac{P(\mathrm{A}|\mathrm{B})\,P(\mathrm{B})}{P(\mathrm{A})} \qquad (6.1)$$

In the above equation, $P(\mathrm{A}|\mathrm{B})$ is the probability of "A" knowing "B". In BAHM terms, the Bayesian formula can be written as:

$$P(TopicMastered \mid questionCorrect) = \frac{P(questionCorrect \mid TopicMastered)\,P(Mastered)}{P(questionCorrect)} \qquad (6.2)$$

However, this equation is far too simple. In BAHM, we test the students by asking them to do the actions inside a virtual environment, with multiple levels of difficulty, although we have another type of questions – multiple choice – but they are few. Clearly getting a hard test correct with many distracters should entail more reward than getting an easy test correct. We therefore substitute *P(questionCorrect)* in equation 6.2 with *P(question of type x of level of difficulty y correct)*. We do a similar substitution for the conditional probability in the numerator.

Doing this substitution, however, leads to too many probabilities that we need to determine. We would need to decide the probability that a simple test of level 0 is correct, the probability that a hard question of level 1 is correct. Deciding on the conditional probabilities does not seem to be an impossible task. However, the non-conditional

probabilities are not very straightforward. How does one decide the probability of a question being correct, without any information about the student's ability?

We need a better way of expressing equation 6.2. If we rewrite equation 6.2 using an equivalent, we can bypass this problem. The new equation, equivalent to 6.2 is:

$$P(Mastered \mid questionCorrect) = \frac{P(questionCorrect \mid Mastered)P(Mastered)}{P(questionCorrect \mid Mastered)P(Mastered) + P(questionCorrect \mid !Mastered)P(!Mastered)} \quad (6.3)$$

(Note: the symbol "!" in the equation stands for negation)

As in equation 6.2, we replace any instance of "question correct" with "question type x of level of difficulty y correct".

Using equation 6.3, we would be required to determine 8 probabilities (one each for question type × level of difficulty × mastered or not mastered). However, this is still too many probabilities to "make up".

We can simplify some of the problems (multiple choice questions) by considering the P(correct | Not mastered) to be the guess factor, which is a function of the level of difficulty and the number of distracters. This can simply be expressed as:

$$\frac{1}{(number\ of\ distracters \times lodfactor)} \quad (6.4)$$

For lod of 0, we are using 0,6 as the lodfactor, and for lod of 1, we are using a lodfactor of 1 (experiential choice).

As a result of equation 6.4, we now only need to create 4 probabilities – 1 probability for a question being correct given the student has mastered the topic for each question type and level of difficulty. The probabilities we are using are empirical and are as follows:

- *P(multiple choice lod 0 correct | mastered) = 0,95*
- *P(multiple choice lod 1 correct | mastered) = 0,89*
- *P(Doing the action lod 0 correct | mastered) = 0,91*
- *P(Doing the action lod 1 correct | mastered) = 0,85*

These probabilities indicate the relative difficulty of each kind of test. We expect that students will have a higher probability of answering correctly an easy *multiple choice* question as they will on hard "*Doing the Action*"questions.

We only need these equations and probabilities tables, since the probability of getting a question incorrect is simply 1 minus the probability of getting a question correct. This holds for the conditional probabilities as well. (e.g. *P(multiple choice lod 0 incorrect | mastered)* = 1 − *P(multiple choice lod 0 correct | mastered)*)

6.3 Study performance

The main interaction that students have with BAHM is through an interactive virtual environment. Therefore, we need a way to judge how much comprehensions the student has gained through these activities.

The Interbook [Brusilovsky, Schwarz 1997], updates the student model by using information about whether a user reads a page or not. However, that system simply records if a page has been displayed, without considering if the page was in fact read. In BAHM, we attempt to determine if a student understands the material that is displayed.

Because we have divided topics into various content objects, the problem essentially becomes judging how sufficiently those objects have been studied. We are using both the time spent and the use of animation help to judge if a content object is sufficiently studied. In general, if a page's associated animation is played, the student will gain more information than if just the text is read. However, there is still no indication that

the student has understood the information presented either in the text or in the video movie.

If the video movie (or animation) has not been played, then we need another way of judging how well the material has been studied. One way to do so is to consider the time spent on each content object. Spending too little time or too much time, reading through a page or doing the actions inside the virtual environment implies that full comprehension has not been reached. However, these timing data can be extremely inaccurate and should not be relied on heavily. For example, a student may leave the computer and not actually look at the video or page. Thus, we cannot distinguish a student who actually spends an hour studying a page from a student who has taken a break for an hour.

Whether the content is sufficiently studied does not warrant a yes or no answer; rather the question is impacted by exactly how much time is spent studying the subtopic. The studied rating on each topic ranges from 0 to 1. If the animation movie for a topic is played, then the rating for that topic is the percentage of the movie that was played.

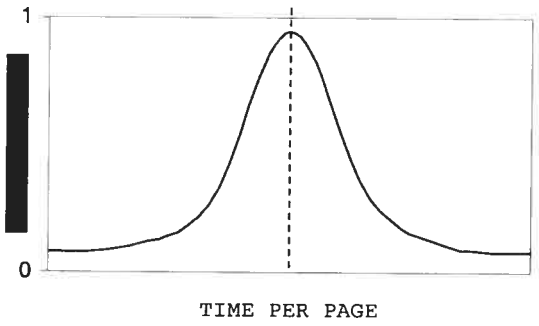


Figure 6.1 How studied values are updated based on time read

This assumes that watching the animation leads to complete understanding, which may not be true. We are looking for alternative measures for judging comprehension in this case.

If the animation is not played, then the rating is dependent on how much time is spent reading the material or doing the action. Currently, the method we use for rating the amount of time spent is to have an average time per page of material that a student should

spend. The planner then multiplies this average time by the number of pages of the content object. If the student spends this average amount, then rating for the object is 1. As she moves away from this point, her score decreases. Of course, even with this method, we cannot be certain if the student understands the material.

To compute a study rating for a topic, the planner sums up the scores for each of the content objects in that topic, and then divides by the total number of content objects for that topic. It should be noted that the content objects belonging to a subtopic are also used to grade that subtopic's ancestors.

6.4 Reviewed Score

The reviewed score on a topic records how many times the student has returned to visit the same topic again. We are interested in recording how frequently a student feels she must review previous material. In general, if she is reviewing frequently, then, she is not retaining information sufficiently, and thus she has not learned the material. Of course, frequent reviewing might reflect individual differences, which currently we do not take into account. The review score records how many times the student has reviewed, as well as how much of the material she views each time.

The overall review score is composed of two factors: a current time score and an overall score. The overall score starts at 0,1 for each topic. Each time the student reviews the topic, the current time's reviewed score is calculated using the method given in section 6.3, using only the current time as the input. This value is then need to adjust the overall reviewed score using the rules in Table 6.1

Table 6-1 Updating rules for review

Current time score	Increase overall value by
Greater than 0,8	50%
Between 0,5 and 0,8	35%
Between 0,3 and 0,5	20%
Less than 0,3	10%

6.5 Combination of these three scores

These three scores, test performance, study performance and reviewed topics, are very hard to reason about individually, and none alone gives a full picture of the student's knowledge. Therefore, we combine the three scores into a single value, indicating how well the topic is "learned".

The most important score on a topic is the test score. If a student has a reasonably high test grade (say over 0,8), then she has demonstrated sufficient mastery of the topic, and the other scores do not matter as much. However, if her test score is not sufficiently high, the other factors become important, and must be considered when calculating the "learned" score. In case the test score of the student is less than 0,8, we combine the three scores (tested, studied, and reviewed) to achieve the learned score.

A simple way to combine the scores is to compute a weighted average, such as:

$$R(learned) = 0,75 \times R(tested) + 0,25 \times R(Studied) - 0,25 \times R(Reviewed) \qquad (6.5)$$

This formula gives more weight to the test scores. However, even if the tests are not required and students do not take them, it is still possible to create a "learned" score on a topic, but the score cannot be higher than 0,25. The tested score can be taken directly from the pre-test. It should be noted, though, that the only way to have a high-learned score on a topic is to perform well on the pre-test or on the tests; direct evidence of understanding is required.

There is one exception to using equation 6.5. If the student has answered all the tests on a topic correctly, then R(learned) is simply 1, i.e. the topic is mastered.

Table 6.2 demonstrates how the "learned" rule can be applied to three students.

Table 6-2 Learned values fro three students

Student	Tested	Studied	Reviewed	Total
1	0,90	0,90	0,0	0,90
2	0,50	0,80	0,70	0,4
3	0,30	0,40	0,15	0,2875

6.6 The evaluation

In this section we explain the evaluation of our system based on the domain "Excendia". The domain for the first course developed with BAHM is Excendia Software and its components. The content for this course is taken from the company's documents at Excendia Inc, Montreal, Canada.

6.6.1 Design

In this section, we discuss the design of the evaluation, including the hypotheses to be tested and the methods for testing them.

6.6.1.1 The hypotheses

Since the system was evaluated at the company, there were a number of main hypotheses that we planned on testing in this evaluation. We expected that users who had the material presented in a way that they could learn by doing, would need less time to learn and would have a more enjoyable experience. Those who used the system without student modeling and learned by reading the material would need more time to read it in detail and would have a less enjoyable experience since it would be up to them to customize the course to get the most out of it. We planned to investigate the time users spent on the course to determine if it is reasonable to use the time spent studying as a component of the student model.

6.6.1.2 Experimental design

Because of confidentiality issue at the Company, we were limited in the number of the users we could obtain for the trial. We chose to design the experiment to be a *within group experiment* [Judd, Kenny, McClelland 2001]. This means that we intended for each participant to be in both the control group and the experimental group. The alternative to this kind of study is a *between group study* in which the members of the control group and

the experimental group are distinct. The advantage of a within group study is the ability to double the numbers for each condition without having to double the total number of participants.

To conduct the experiment, we divided the participants into two groups – the first group had the aid of the student model and virtual environment to learn by doing for the first half of the course and the second had the aid of the student model during the second half of the course. Since the users could traverse the course non-linearly, we could not define halves of the course by which topics are studied. Rather, we defined halves by the number of topics studied.

In this experiment, the student model and virtual environment were used for both adaptive navigation and adaptive presentation. In order to determine the impact of the student model and our virtual environment on our hypotheses, we planned to survey users in the middle of the course and at the end of the course. These surveys would provide subjective measures on how the users found the system.

In terms of improved facilitated learning, we did not expect much difference in test performance on concepts, since even without the aid of a student model and a virtual environment, users could still alter the course. However, if they did not do this, then we expected those users who performed the actions inside the VE to perform better on the tests.

6.6.1.3 Results

The users did not go as originally intended. Only one user completed the course, and no other users even made it to the midway point. Therefore, we were not able to perform a within group study. Because of these difficulties, we were not able to look at hypotheses that compared the differences between the experimental group (with the student model and VE) and the control group (without the student model and VE). The hypotheses we could test concerned the relationship between times studied and test performance. Considering that the pre-test's grades could not have any effect on these results, we believe that pre-test's grade can only be the starting point for the users.

We present the results of this study as a case analysis. By this we mean that we present results for each student, rather than combine the data from all students into a single population and report the results for the population.

6.6.1.4 Adaptation through user's instances

The default algorithm for classifying objects is the one presented in section 5.6.2.2. With this algorithm, only those objects at the correct level of difficulty are classified, and others are assumed to be not wanted. This default algorithm essentially predicts, by default, that objects at an incorrect level of difficulty will not be wanted. In order to determine if this assumption is correct, we counted the accuracy of predictions for both sets of objects for determining the algorithm's overall accuracy. Table 6.3 presents the accuracy using this default algorithm.

For this algorithm, the Naïve Bayes Classifier simply determines if the probability that an object will be wanted is greater than the probability it will not be wanted. This translates to a 50% threshold for determining whether or not to show an object. However, changing this threshold may change the accuracy. The result of using different thresholds is shown in table 6.4.

Table 6-3. Accuracy of classifier when objects at the right level of difficulty and assuming objects at the wrong level of difficulty will not be wanted

student	accuracy
1	84,11
2	74,16
3	60,06
4	76,97
5	84,71
6	59,50
7	84,62
8	90,53

Table 6-4. using different thresholds when classifying objects at the correct level of difficulty and assuming others to be not wanted

student	30%	70%	90%	Max t from 50%
1	63,08	81,78	79,44	-
2	67,03	79,69	80,20	6,254
3	60,06	64,09	61,86	3,716
4	49,34	76,97	76,97	-
5	69,41	78,82	78,82	-
6	60,25	58,25	58,50	1,343
7	81,54	84,62	86,15	1,00
8	81,05	91,58	92,63	1,422

From table 6.4, we can see that changing the threshold for the Naïve Bayes Classifier did affect accuracy, improving the results for five of the eight students. However, this default algorithm assumes all objects at an incorrect level of difficulty will not be wanted.

6.6.1.5 Timing and grading information

We are using timing information for grading topics and concepts as well as student's performance. However, we are not sure if this is a reasonable thing to do, i.e. does timing information provide the tutor any information about whether or not a student has learned the material? Or as whole, is this method, (learning-by-doing) more effective than simply reading the material? To determine this, we analyzed how much time students spent on a topic and their test performance on those topics. We did not perform the same analysis for concepts since if a student studies one item in a concept, she will be given the tests on the concept at her level of difficulty, possibly including the question on things she may not have seen. This is not the case for topics.

To determine if timing information provide any insight into student knowledge, we correlated how much time a student spent studying a topic with how she performed on the test. We ignored the time the student has spent on answering the test. How well learners explored exercises did not seem to depend on time spent in each exercise but on how long they spent reasoning about their actions and how long they spent exploring each individual exploration case [Bunt, Conati 2003]. We also ignored the level of difficulty of questions for this analysis and simply looked at the overall percentage of questions correctly answered. The results are given in table 6.5.

Table 6-5 Correlations of time spent studying and test performance

Student	Correlation	Significance of two-tailed t-test
1	− 0,353	0,117
2	− 0,278	0,029
3	− 0,566	0,001
4	0,602	0,038
5	− 0,209	0,619
6	− 0,0105	0,668
7	0,798	0,018
8	0,355	0,348

From table 6.5, we see that for half of the students, there was a significant correlation between times studied and test performance. Furthermore, we see that for five students, there was a negative correlation while for three students, there was a positive correlation. A negative correlation means that the less time the student studied a topic, the higher her test grade. A positive correlation means just the opposite – the more time studying a topic, the better the student did on the tests.

While we do not have enough data to draw any firm conclusions, we can use it as a hint for future research. For half of the students, there was a significant correlation between time studied and test performance. However, the direction of the correlation was not the same for each student. For some students, studying longer through reading the text

or seeing the images meant better scores while for others, studying less time and watching the video and practicing inside the VE meant better scores. Therefore, having one grading policy to cover all students, such as "the longer the student studies the material, the less likely she is to have learned it" is flawed. This preliminary data shows that we must treat each student individually and attempt to learn those individual differences in order to create an accurate student model. However, we are not only looking for higher grades in tests, but we expect that this method of learning is cost effective for the company.

Furthermore, we must also incorporate these differences in teaching policies, since trying to speed up a student who does better when she is allowed more time studying and reading the material would not be beneficial to that student, but in most cases speeding up the learning process would help the company save money as it happens at Excendia. Systems that treat all students the same are missing a great opportunity to truly individualize the learning experience.

Chapter VII

7. Conclusions

> ***"Mostly e-learning has concentrated on the transmission of knowledge, whereas most learning in business is skills-based. Skills benefit from 'learning by doing' techniques, which means simulations and games techniques."*** – Donald Clark CEO of leading European provider of e-learning solutions, Epic plc.

In this thesis, we presented work on BAHM, a web-based adaptive hypermedia and interactive multimedia system for presenting lectures and enabling the students to learn by doing inside a virtual environment. We discussed the domain organization, the student model, and evaluation of a course with eight users.

Both the virtual environment construction and the evaluation posed significant difficulties. We started from preexisting linear course material and converted it to the BAHM course structure. We also created the animation help which was time consuming. It was not easy to create a non-linear course from an already defined structure. In terms of the evaluation, it was pretty hard to ensure completion of the course. Performing a real world evaluation turned out to be quite easier said than done an undertaking.

In BAHM, the domain model consists of four main components – topics, concept, content objects, and tests. Topics were arranged in a semantic network with link types prerequisite, co-requisite, related, and remedial. The topic structure was recursive – topics have subtopics which are also topics and so forth.

Concepts were keywords that appear throughout the course. They did not have content dedicated to their teaching explicitly. Rather, whenever they appear in the topic material, the planner could take the opportunity to teach them at that point, based on the user's knowledge and preferences. Both topics and concepts were composed of content objects. There were three kinds of content objects – text, graphics, and multimedia (help-type, test-type). These objects were the actual content the student sees. Tests, instead of being true/false, were practical quiz for which student had to find the solution by doing inside a virtual environment. The tests were so designed, as if they were real world actions and users felt themselves inside the real environment. Tests covered either topics or concepts or both.

The student model was used for personalizing the course through two means: adaptive navigation and adaptive content. Adaptive navigation helps guide students through a non-linear course. The student model is used to either pick the next topic for the student to study or provide advice if the student chooses the topic herself. Topics are graded based on how well they have been studied and how the student has performed on a topic's pre-topics.

The contribution we made to the area of adaptive navigation was to consider not just the test grades and whether a page was "seen" or not. We also considered how a student studied the course material to determine if there was relationship to the test grades. Although we have only preliminary data in this area, we see that there are individual differences in time spent studying and test performance. Thus it seems a promising area for future research.

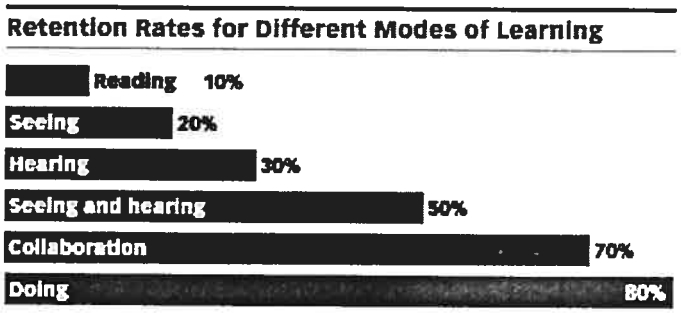
Adaptive content and link change dynamically. This was accomplished through a two phase process. The first took into consideration how well the student understood the material and chose content at the right level of difficulty. The second incorporated a student's preferences. These preferences were learned by the planner by observing the kind of content the student elected to see in the past. A Naïve Bayes Classifier was used to learn these preferences.

We performed on evaluation of BAHM with eight users at Excendia Company. We tested how accurately the Classifier could predict what objects the user would elect to see, such as image, text, examples, definitions or video animation. We also could see the progress of the users when they learned by doing instead of learning by reading the text. We could see that the students who practiced inside the virtual environment spent less

time to complete the action. Using Naïve Bayes algorithm and adding some enhancements such as changing the decision thresholds, we were able to achieve a minimum of 63% accuracy for each user. This means that the classifier could accurately predict actions the user will take at least 63% of the times.

Five users out of eight spent less time studying, meaning practicing inside the virtual environment and learning by doing, and received high grades in the tests. We see that in most of our cases people learned better by doing and getting involved in solving the problem instead of reading the material theoretically. A more thorough experiment would be required to draw general conclusions. Spending less time on training has direct impact on costs for the companies. Furthermore, we see that using the virtual environment not only reduces the risk of training, but improves the features to deliver complex content to trainees.

Chi, Bassok, Lewis, Reimann, and Glasser also studied the fact that learning-by-doing results better than the other methods and their result is shown in figure 9.1.



Figure[10] 7.1 Retention rates for different modes of Learning

Although, this study did not involve a large number of users, we see promising results using the system. This study indicates that there is not a consistent correlation between the amount of time spent studying and a student's performance on the tests, contradicting most people belief. For some users, studying for a longer time produced better test results, while for others, studying for a shorter time meant better results. It somehow depends on the methods users choose to study. Users have individual

[10] Source: eMarketer, Inc. 2003.

differences, so that we must have multiple teaching policies that are appropriate for different kinds of users.

Furthermore, systems must be able to learn which of these multiple teaching policies to use for each student, and they must be able to adapt their decisions as students use the system. Students may change how they interact with the system as they become more comfortable with it. Furthermore, their knowledge and abilities change while using the system. Students do not remain static while learning, therefore, the system should also not be static while the student is learning; it should be able change and adapt just as the students change and adapt. The system should always keep the students motivated by making them feel they are solving a real world problem.

Appendix A. Regression equations

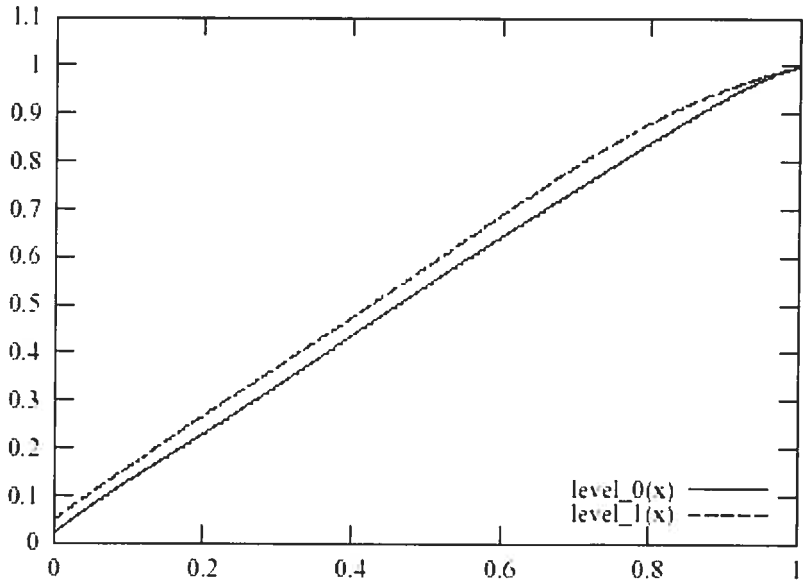


Figure A 1 Regression equations for updating the level 0 mastery when the student did not study objects for enough time

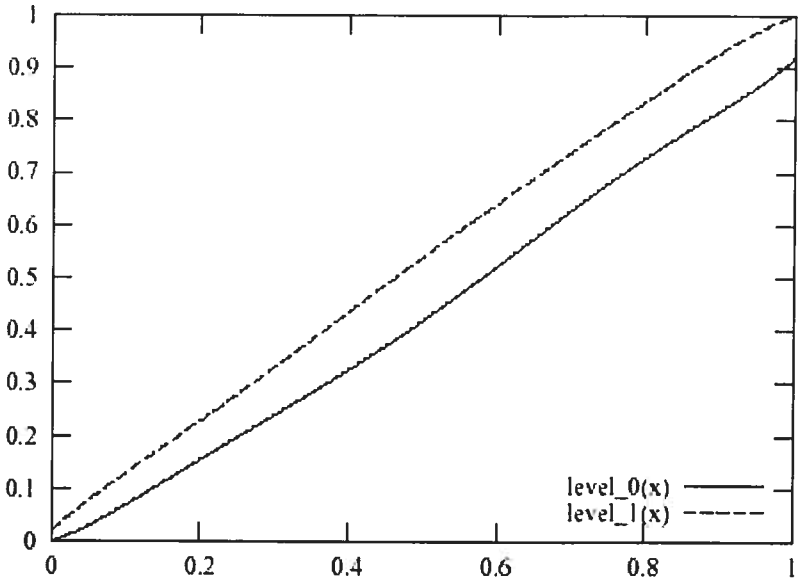


Figure A 2. Regression equations for updating the level 1 mastery when the student did not study objects for enough time

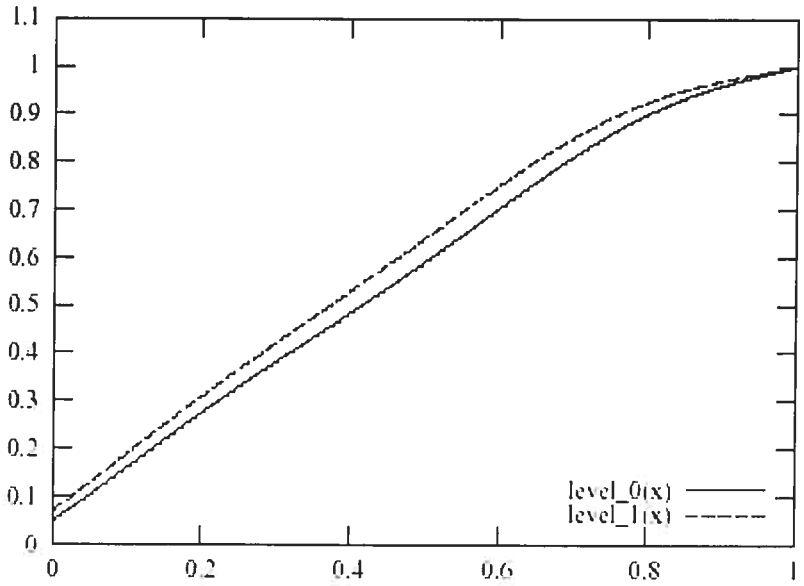


Figure A 3. Regression equations for updating the level 0 mastery when the student studied objects for an average amount of time

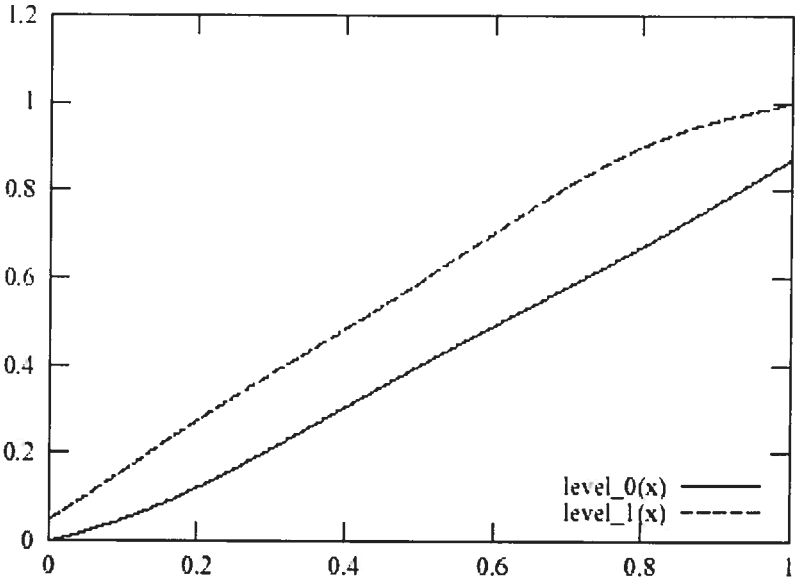


Figure A 4. Regression equations for updating the level 1 mastery when the student studied objects for an average amount of time

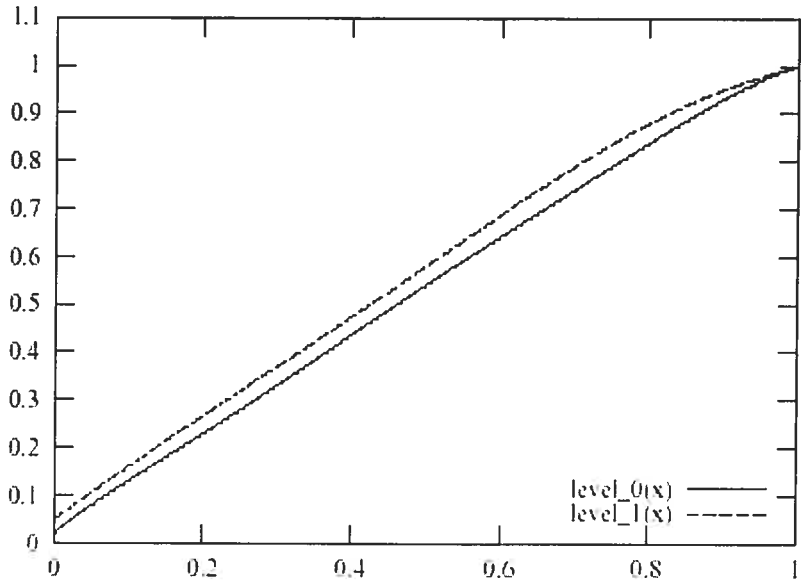


Figure A 5. Regression equations for updating the level 0 mastery when student studied objects for too much time

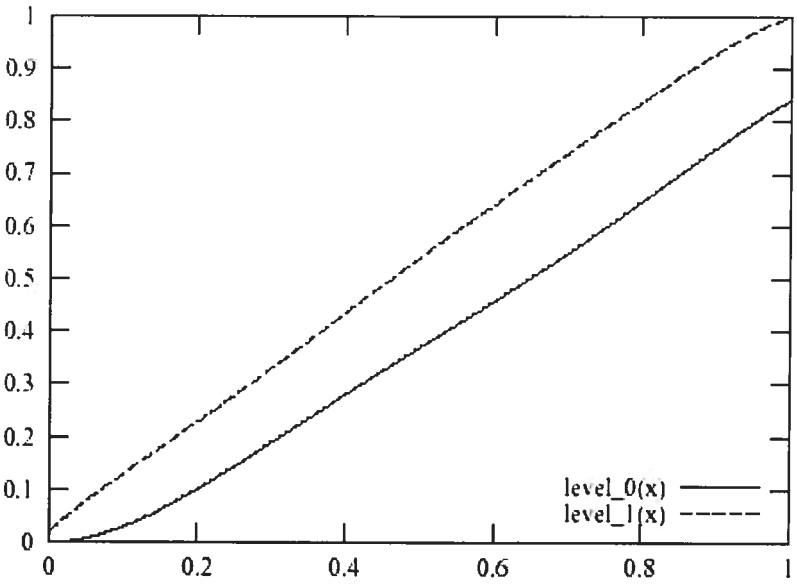


Figure A 6. Regression equations for updating the level 1 mastery when the student studied objects for too much time

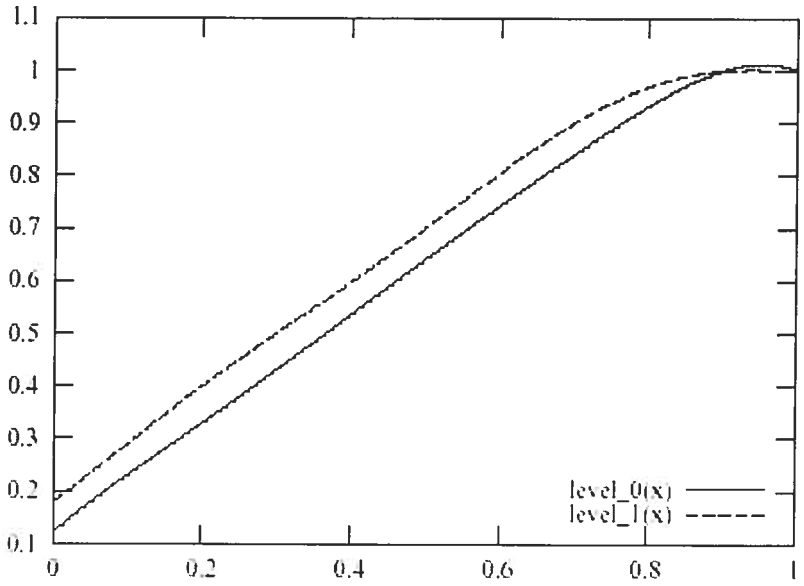


Figure A 7. Regression equations for updating the level 0 mastery when dong the action with no errors

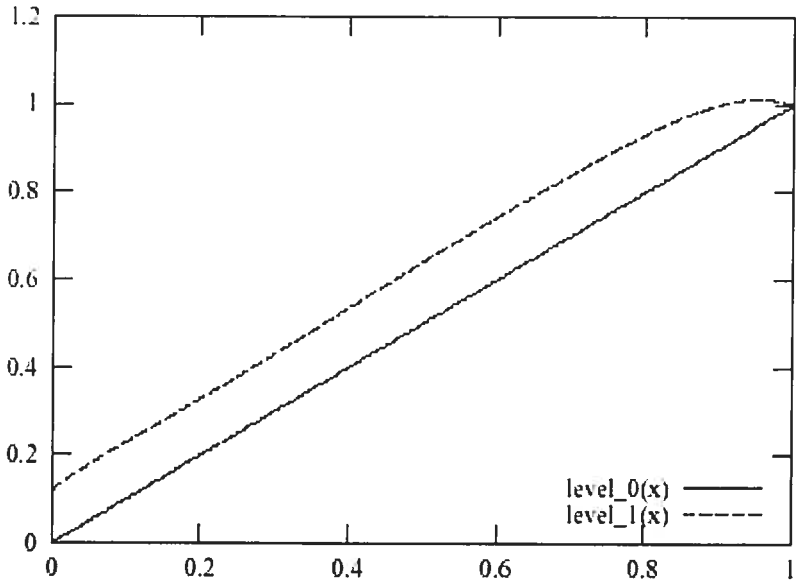


Figure A 8. Regression equations for updating the level 1 mastery when dong the action with no errors

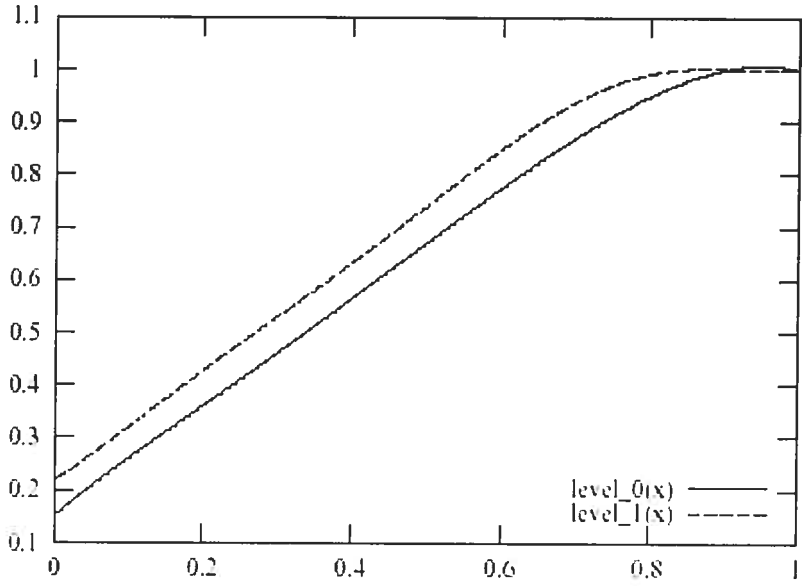


Figure A 9. Regression equations for updating the level 0 mastery when answering 4-answer multiple choice question correct

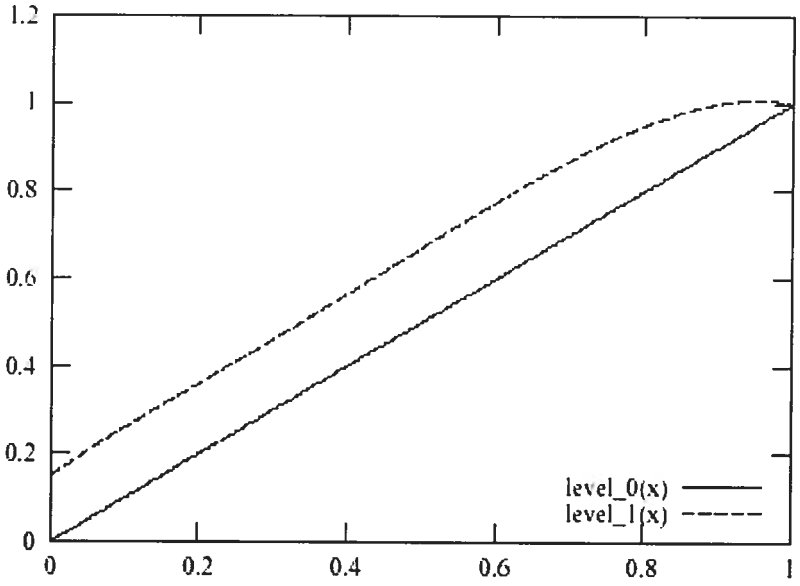


Figure A 10. Regression equations for updating the level 1 mastery when answering 4-answer multiple choice question correct

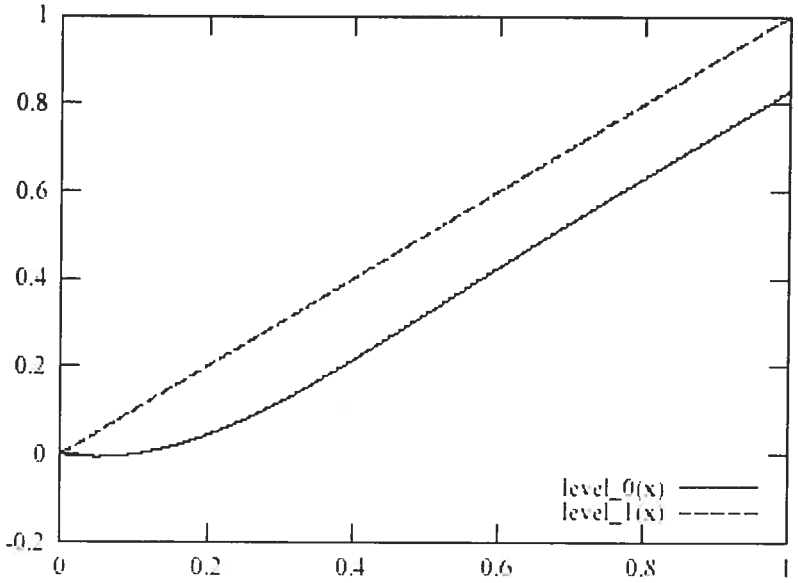


Figure A 11. Regression equations for updating the level 0 mastery when doing the action with errors

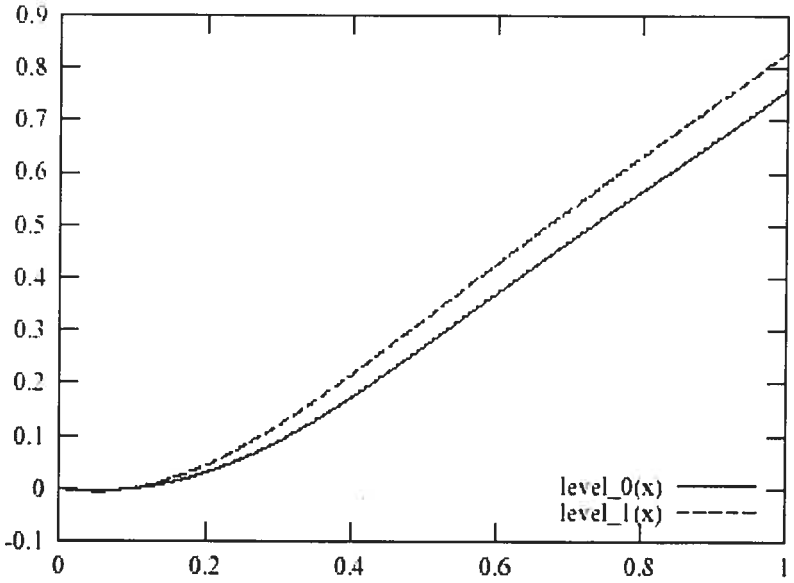


Figure A 12. Regression equations for updating the level 1 mastery when doing the action with errors

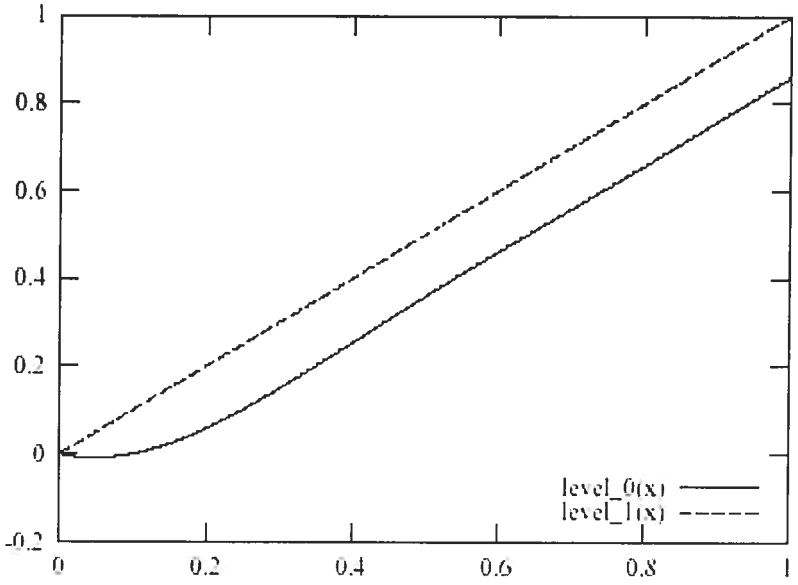


Figure A 13. Regression equations for the updating the level 0 mastery when answering 4-answer multiple choice equation incorrect

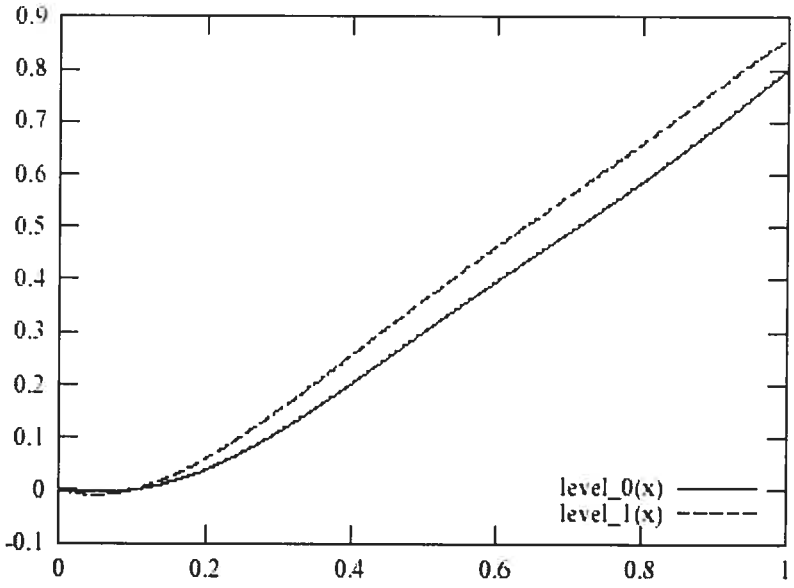


Figure A 14. Regression equations for the updating the level 1 mastery when answering 4-answer multiple choice equation incorrect

Appendix B. BAHM Application domain - Excendia

In this section, we discuss about the application domain which we applied the BAHM system. We also take a detailed look at the components of Excendia and its functionalities.

Excendia telephony application domain

Excendia Office Telephony® (OT) is an integrated suite of communications and customer interaction applications and development tools developed by Excendia Inc. http://www.excendia.com

Application Modules

The Office Telephony suite is composed of tightly integrated and interoperable application modules. These include telephony applications that can run on the Intel Converged Communications Platform to process calls and provide contact centre agents with desktop tools to manage their calls.

OT FrontDesk

An automated attendant and call switching application that answers and forwards calls automatically. *FrontDesk* greets callers in accordance with pre-administered profiles that specify appropriate language and routing instructions. It includes ACD and calls queuing functionality and can be used in tandem with a speech recognition system. *FrontDesk* generates detailed call records for each call it handles.

OT UniMail

A unified messaging module that enables users or contact centre agents to access and manage their voice, fax, and email messages from their desk using Microsoft Outlook, or from outside the company using a telephone or Web browser.

OT TeleCalendar

An appointment scheduling and management application that lets callers schedule, review, and manage appointments from anywhere. Using Microsoft Outlook or a telephone, *TeleCalendar* users can review, accept, or reject their appointments as well as indicate the time periods during which they are available. *TeleCalendar* also reminds callers of their appointments and notifies them if any change occurs.

OT HelpDesk

A contact centre application that can be customized to meet the customer service or technical support needs of any business. *HelpDesk* identifies callers for intelligent routing and automatic access to customer records via screen pops. *HelpDesk* can be combined with Customer Relation Management (CRM) software to provide intelligent customer contact solutions.

OT SoftPhone

A desktop graphical tool that enables contact centre agents to control their calls and manage other contact centre functions. *Softphone* can be configured to appear whenever a call arrives, displaying the caller's name and/or telephone number. Agents can view call

queues and logs, auto dial from a contact list, and change their work status so that their calls will always be handled appropriately.

OT Manager

A desktop graphical tool for Office Telephony administrators; OT Manager is used to configure the system (add or delete agents, add voice mailboxes, etc.) and provide system-monitoring functions.

Intel® Converged Communications Platform

Excendia Office Telephony software is available for the Intel® Converged Communications Platform, which consists of a 2U rack-mountable server integrated with IP and PSTN boards, Windows® 2000, Intel® NetMerge™ converged communications server software, and management software. Each node supports between 16-48 stations. The Intel Converged Communications Platform supports IP, and analog and digital trunks and stations.

Intel® NetMerge™ Converged Communications Server Software

Intel NetMerge converged communications server software is ideal for designing open, standards-based telecommunications servers that support computer telephony (CT) applications, such as messaging, fax, automatic call distribution, and other applications from software and hardware developers. Intel is committed to making Intel NetMerge converged communications server software the centerpiece for enabling next-generation converged voice and data solutions.

Where Will Excendia Get Physically Installed in a Company?

Excendia Virtual Assistant solution can be installed as an application in a standalone "all-in-one" converged communication phone system, or as an adjunct media server

controlling the existing PBX. The decision to go with a standalone converged system or adjunct media server depends on the size and investment already made in the existing PBX as well as the readiness to embrace the new open converged communication technology. A third alternative would be to install a hybrid configuration where Excendia controls PBX extensions as well as new converged stations (IP and non-IP stations). This hybrid configuration provides a smooth transition path from legacy PBX's to open converged communications systems.

In all cases, Excendia sits physically between the information server (Microsoft Exchange) and the telephone system (as adjunct to the PBX or inside the PCX) (figure 6.3). In this position, Excendia uses the corporate data to add "intelligence" to the phone system, and uses the phone system to provide voice-access to data stored in the Exchange server (Inbox, Calendar, Contacts, and Tasks).

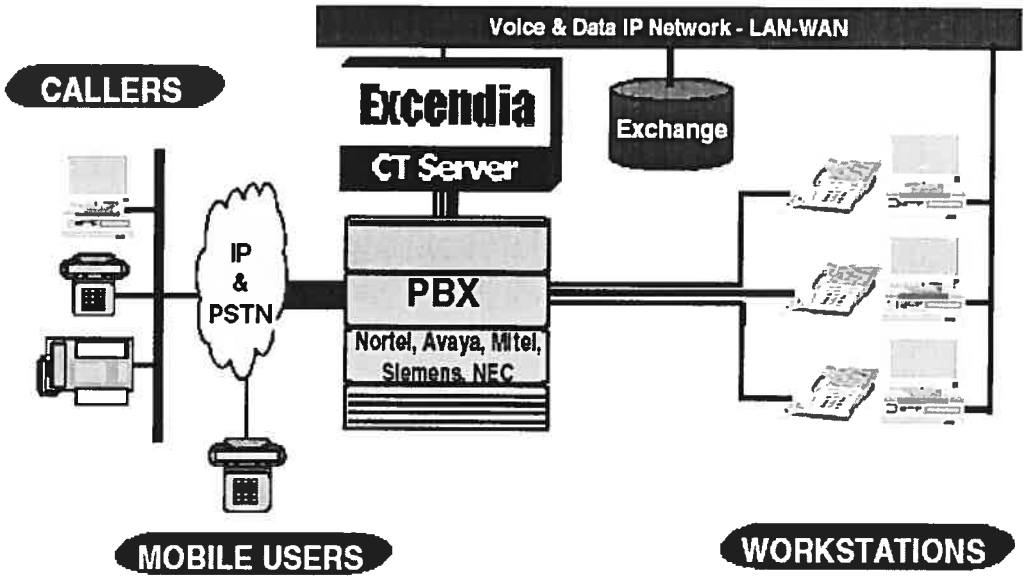


Figure B 1 Excendia as an Adjunct Media Server to existing PBXs

Access & Process All the User's Mail by Phone:

Excendia reads the e-mails to the user over the phone and lets user respond to them verbally. It can forward the user's faxes anywhere needed them. It enables the user to forward any voice mail or record a new voice message and send it as e-mail to any contact in the user's address book. That's because Outlook-by-Phone is tightly integrated with the company's Exchange server and telephone system. User can retrieve all her voice mail, faxes and e-mail from any phone while traveling or visiting customers.

- It can access your own Outlook account and read your e-mails to you over the phone and even let you respond verbally. In multilingual environments, Outlook-by-Phone will automatically use the appropriate language to synthesize e-mails and read them to you correctly.
- It enables the user to call back voice mail senders or forward a voice message as e-mail to any contact in your address book.
- You can also record a new voicemail over the telephone and send it as a sound e-mail to anyone in the user's contact database.
- It can send the faxes anywhere the user needs them. It can also forward them as an e-mail to anyone of the user's contacts;

And because your voice and fax communications are treated by Outlook as e-mail messages, user has access to the rich set of Outlook functions to manage them. She can for instance organize all faxes, e-mails and voice messages associated with a particular project or client in a single electronic folder on her computer.

Access & Manage the Voice and Faxes from Outlook©

It enables the user to view and print her faxes from her Outlook© Inbox. Since the voice messages and fax documents are treated by Outlook© as e-mail messages, she can forward them as e-mails to any contact in her address book, or organize them in electronic folders grouped by project or client.

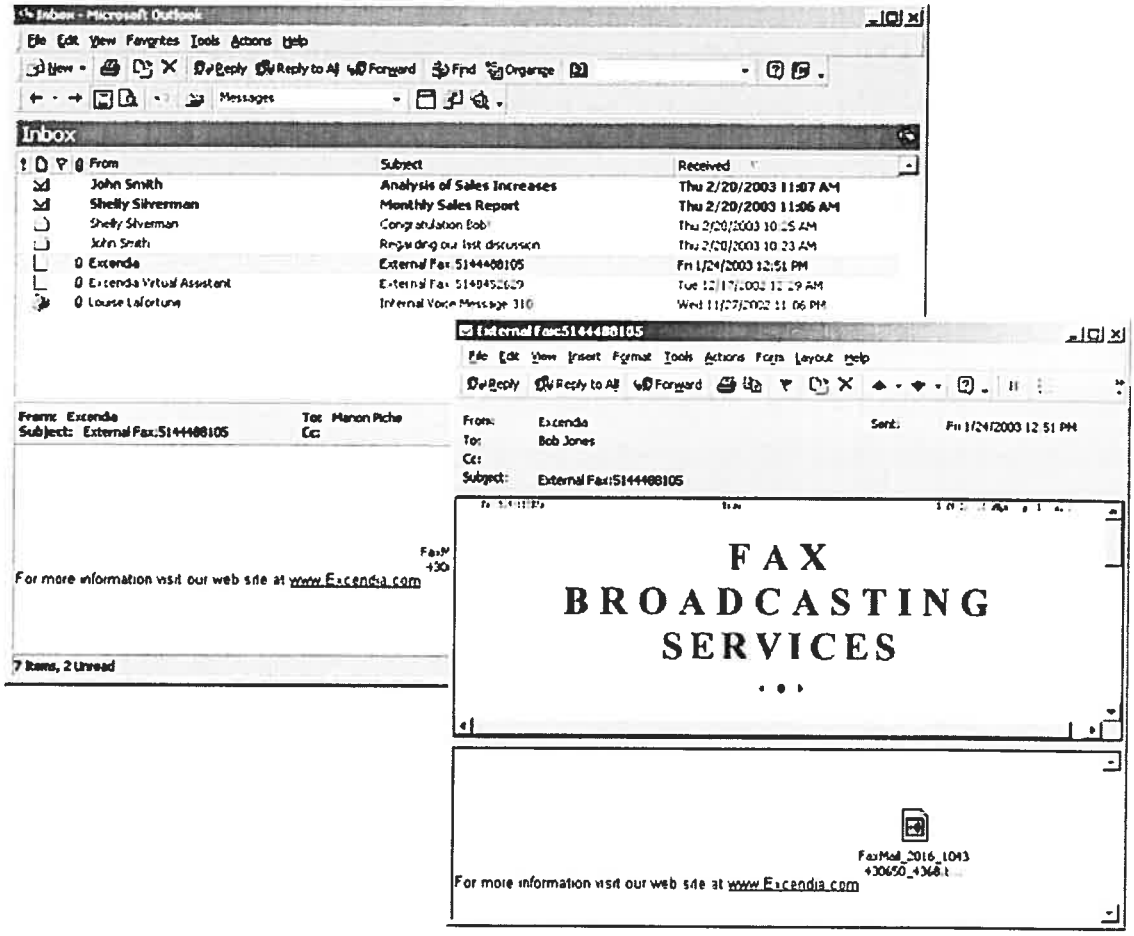


Figure B 2. Access & Manage the Fax from Outlook

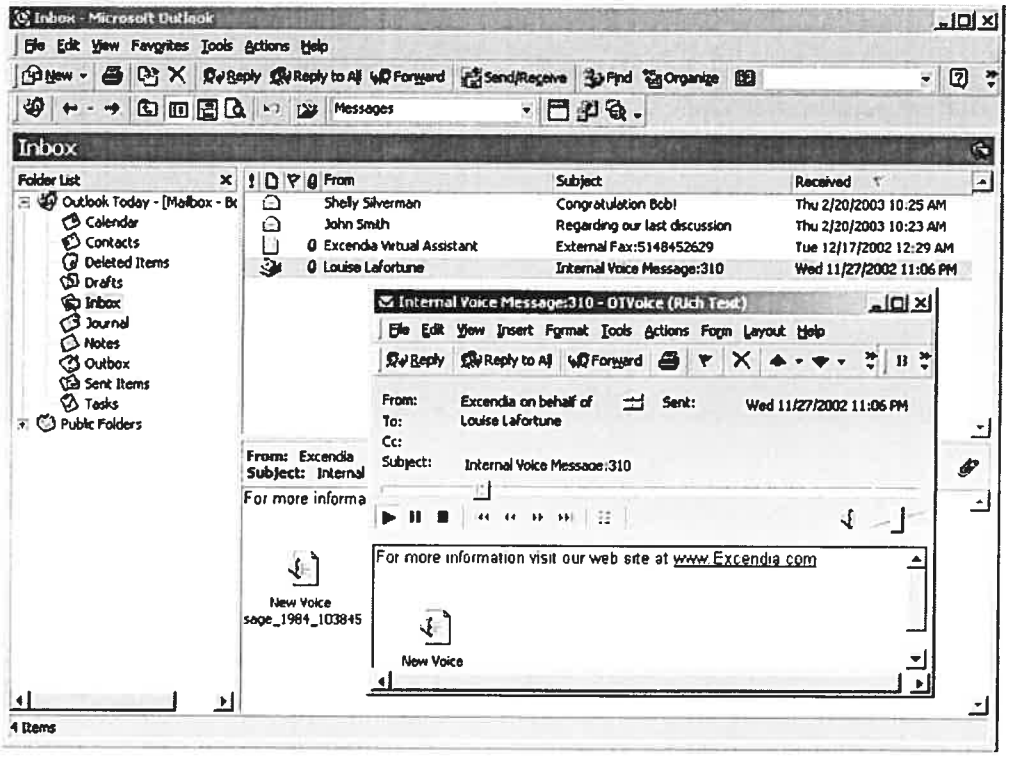


Figure B 3. Access & Manage the Voice from Outlook

Access & Manage the Appointments by Phone

Each time the user calls, Excendia will tell her about any meetings and appointments she has for any date she specifies. And for each appointment, it gives her the list of attendees, the place of meeting and the topics to be discussed. User can even organize meetings by phone. She simply specifies when, where and who should attend, and Excendia calls the meeting on her behalf.

- User can access her Outlook Calendar and listen to her meetings and appointments for any date.
- User can also schedule new meetings by phone. Use her own voice to define the meeting agenda, and let Outlook-by-Phone e-mail the invitation to all attendees, as if the meeting was organized from her desktop.

Access & Communicate with the Contacts by Phone

Using the voice commands or touch-tones, user can select a contact from her address book, and have Excendia call and connect her. The new generation of mobility solutions that does not require the user to carry any special communication device such as a PDA or a Blackberry. Any wireless or public phone can be used to communicate with the contacts because the information and communication intelligence reside on the corporate server, not the terminal device.

Access & Manage the daily Tasks by Phone

Excendia Virtual Assistant reminds the user about her to-do tasks and their current status each time the user calls. User will no longer forget to call someone, or to complete something. User can even record a new to-do task from her phone and have it added to her Outlook© Tasks list.

Executive Contact Center

As the personal assistant, Excendia greets the callers by their name and in their language, before transferring them to the user, and if the user is not at her desk, it locates her and forwards her the calls based on her instructions. And when it transfers the calls, it will let user know who is calling so user can decide to take the call, leave it in call queue, or forward it to an associate or to voice mail (figure 6.5). It also pops up the caller contact information on the screen when user is at your desk. Excendia will keep a log of all incoming and outgoing calls for user's future reference and easy call back. It will also keep a separate log of the various interactions with each contact.

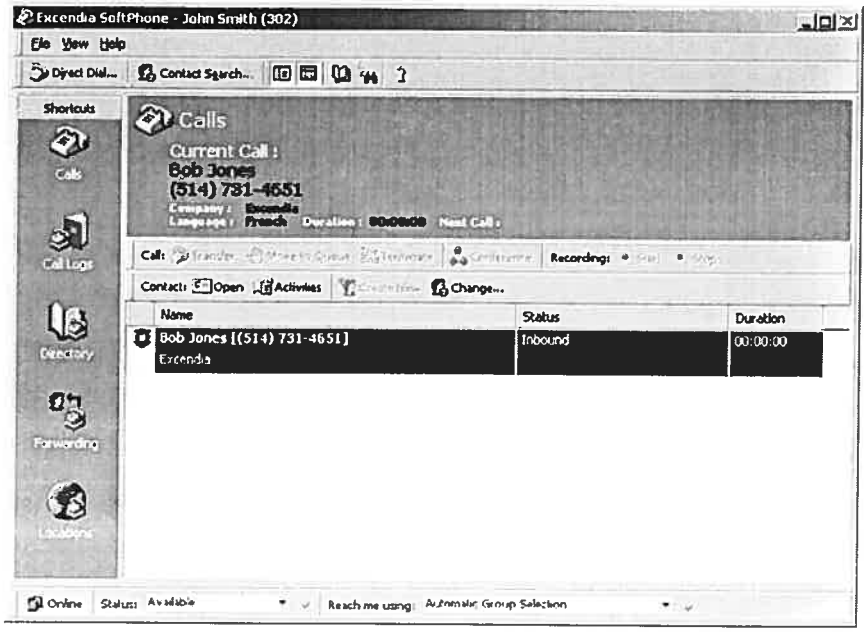


Figure B 4. View the incoming call

Intelligent call routing

Excendia FollowMe is a standards-based, intelligent call routing application that makes the user always reachable for important business calls. Calls follow the user only when and where she wants to take such important calls. User sets the rules and Excendia makes sure she gets the calls she wants to answer. FollowMe is a schedule-based and user-centric application. It always checks the user's personal call forwarding schedule to decide where to route each call. The date and time of call are used to determine if and where FollowMe will forward incoming calls to the user. The call forwarding actions may include locating the user on different numbers before going to voice mail. For example, the default call-forwarding schedule may include the sequence: "*Ring my extension*, if I don't answer, *call me on my mobile phone*, and if there is no response, *go to my voice mail*". Actions may also include forwarding the calls to colleagues or associates, using their own FollowMe schedule.

Know Who's Calling before Answering the Call

Each time a call is forwarded to user, Excendia FollowMe will let her know who is calling so user has the option to answer the call, transfer it to a colleague or forward it to voice mail.

How Does It Work?

Excendia FollowMe is a software application running on an Excendia Server Platform that either sits behind your telephone system or directly connected to the public telephone network. In other words, Excendia FollowMe uses the phone system or direct telephone lines to forward unanswered calls to different user locations.

Setup & Run Conferences Using Outlook© Calendar

With Excendia Outlook Conferencing, user can set and run multi-party conference calls easily and rapidly using Outlook© Calendar. User can simply schedule the event in Excendia Conference calendar, select the participants from the Contacts, and let Outlook Conferencing do the rest. Excendia will establish a conference bridge and e-mail invitations to participants with all the details on how to join the conference.

Automatic Notification & Reminders

User can instruct Excendia to automatically notify her when certain events happen such as an upcoming appointment or reservation, a new important message, etc. Based on her notification preferences, Excendia will alert her about selected events, at specific time schedules, using her preferred notification methods: phone, SMS, e-mail or fax.

Detailed Call History

Excendia keeps a log of the user's telephone interactions with customers and business associates.

This feature is particularly useful for business follow-up, billing and customer relationship management.

Multilingual & Multi-Tenant Support

Excendia can handle as many languages as the market requires, because it was designed for multilingual environments. And for hosted services, Excendia supports multi-tenant operations by keeping each tenant's environment autonomous and independent from others.

Bibliography

Agogino, Alice M. and Hsi, Sherry, (1995). "Learning Style Based Innovations to Improve Retention of Female Engineering Students in the Synthesis Coalition," In *Proceedings ASEE/IEEE Frontiers in Education Conference*, Vol. 2, pp. 667-670.

Aïmeur E., Salchian B. (2003). A Network-Based Training Environment Based on Kolb's Theory. In *3rd IEEE International Conference on Advanced Learning Technologies ICALT2003.*

Anderson J.R., Farrell R., & Sauers R. (1984). Learning to program in LISP. *Cognitive Science*, 8, pp. 87-129.

Arcos J. L., Muller W., Fuente O., Orùe L., Arroyo E., Leaznibarrutia I., Santander J. (2000) *ITS, 5th International Conference, ITS 2000, Proceedings*. Gilles Gauthier, Claude Frasson, Kurt Vanlehn (Eds) Montreal, Canada. pp. 43-52.

Ardissona L. Console L., Torre I. (1999). Exploiting User Models for Personalizing News Presentations. In *The Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.*

Ardissono L., Goy A. (2000). Dynamic Generation of Adaptive Web Catalogs. In *The Proceedings of Adaptive Hypermedia and Adaptive Web Based Systems*, Italy, pp. 5-16.

Armstrong R. Frietag D. Joachims T. and Mitchell T. (1995). WebWatcher: A Learning Apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering from Heterogeneous Environments*, Stanford, CA.

Arroyo I., Beck J., Beal C. R., & Woolf B. P. (2003). Learning within the Zone of Proximal Development with the Animal Watch intelligent tutoring system. Accepted to the *American Educational Research Association annual meeting*, Chicago IL.

Beaumont I. (1998). User Modeling in the Interactive Anatomy Tutoring System ANATOM-TUTOR. In *Adaptive Hypertext and Hypermedia*, Brusilovsky P. Kobsa A. and Vassileva J. (Eds.), Kluwer Academic Publishers, Netherland, pp. 91-116.

Bental D., Cawsey A., Pearson J., and Jones R. (2000). Adapting Web-Based Information to the Needs of Patients with Cancer. In *The Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, Italy, pp. 27-37.

Birkey, Richard C., and Rodman J. (1995). Adult Learning Styles and Preference for Technology Programs. *National University Research Institute,1995 Lifelong Conference Proceedings*.

Bloom B.S. (1984). The 2 Sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), pp. 4-16.

Bodner R., Chignell M., Tam J. (1997). Website Authoring using Dynamic Hypertext. In *The Proceedings of Webnet*, pp. 104-109.

Bollen J., Heylighen J. (1997). Dynamic and adaptive structuring of the World Wide Web based on user navigation patterns. In *Flexible Hypertext Workshop, help at the Eight ACM International Hypertext Conference*, Sydney, pp. 13-17.

Boyle C., Encarnacion A.O. (1998). MetaDoc, an Adaptive Hypertext Reading System. In *Adaptive Hypertext and hypermedia*, Brusilovsky P. Kobsa A. and Vassileva J. (Eds.), Kluwer Academic Publishers, Netherlands, pp. 71-89.

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

Brusilovsky P. (1998). Methods and Techniques of Adaptive Hypermedia. In *Adaptive Hypertext and Hypermedia*, Brusilovsky P., Kobsa A., and Vassileva J., (Eds.), Kluwer Academic Publishers, The Netherlands, pp. 1-44

Brusilovsky P. (2001). Adaptive hypermedia, User Modeling and User Adapted Interaction, 11 (1/2) pp. 87-110.

Brusilovsky P. and Anderson J. (1998). ACT-R Electronic Bookshelf: An Adaptive System to Support Learning ACT-R on the Web. In *The Proceedings of Webnet*, Toronto, Canada.

Brusilovsky P. Cooper W. (1999). ADAPTS, Adaptive Hypermedia for a Web-based Performance Support System. In *The Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.*

Brusilovsky P. and Eklund J. (1998). A Study of User Model Based Link Annotation in Educational Hypermedia. *Journal of Universal Computer Science 4.*

Brusilovsky P., Pesin L., Zyryanov M. (1993) Towards an adaptive hypermedia component for an intelligent learning environment. In Human-Computer Interaction, *Lecture Notes in Computer Science #753,* Bass L.J., Gornostaev J. and Unger C. (Eds.) Springer-Verlag, Berlin, pp. 348-358.

Brusilovsky P., Schwarz E., and Weber G. (1996) ELM-ART: An intelligent tutoring system on World Wide Web. In *The Proceedings of Intelligent Tutoring Systems, Lecture Notes in Computer Science, Vol. 1086* Frasson C. Gauthier G., and Lesgold A. (Eds.), Springer, Berlin, pp. 261-269.

Brusilovsky P., Schwarz E., and Weber G (1996). A tool for developing adaptive electronic textbooks on WWW. *In Proceedings of WebNet'96 – World Conference of the Web Society*, Boston MA, USA, pp. 165 - 174.

Brusilovsky P., Schwarz E. (1997). User as Student: Towards an Adaptive Interface for Advanced Web-based Applications. In *User Modeling: Proceedings of the sixth International Conference, UM97*, Jameson A., Paris C., and Tasso C. (Eds), Sprinter Wien New York, pp. 177 – 188.

Brusilovsky P., Stock O., and Strapparava C. (2000). Adaptive Hypermedia and Adaptive Web-based Systems, AH2000. *Lecture Notes in Computer Science, Vol. 1892*, Springer-Verlag, Berlin.

Bunt A. and Conati C. (2003). Probabilistic Student Modeling to Improve Exploratory Behavior. *Journal of User Modeling and User-Adapted Interaction,* vol 13 (3), pp. 269-309.

Calvi K., De Bra P. (1997). Improving the Usability of Hypertext Courseware through Adaptive Linking. In *Flexible Hypertext Workshop, help at the Eight ACM International Hypertext Conference*, UK.

Cantor, Jeffrey A. (1992). Delivering Instruction to Adult Learners. Toronto: *Wall & Emerson*. pp. 35-43.

Carbonell J.G. (1970). AI in CAI: An artificial intelligence approach to computer-assisted instruction. In *IEEE Transactions on Man-Machine Systems,* 11, pp. 190-202.

Carro R., Pulido E., and Rodriguez P. (1999). TANGOW, Task-Based Adaptive Learner Guidance on the WWW. In *The Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.*

Cini A., Valdeni de Lima J. (2002). Adaptivity Conditions Evaluation for the User of Hypermedia Presentations Built with AHA. *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer-Verlag, LNCS Vol. 2347, pp. 490-493.

Conati, C., VanLehn, K. (1996) POLA: A Student Modeling Framework for Probabilistic On-line Assessment of Problem Solving Performance. In *Proceedings of the Fifth International Conference on User Modeling*. Kailua-Kona, HI.

Cristea A. and Okamoto T. (2001), Considering automatic educational validation of computerized educational systems, *IEEE International Conference on Advanced Learning Technologies, ICALT2001*, USA.

Cristea A. and De Bra P, (2002). *ODL Education Environments based on Adaptivity and Adaptibility*. Lecture Notes in Artificial Intelligence, 2198, Springer, 2001, 148-152.

Crowley R. and Medvedeva O., (2003). SlideTutor – A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis. Accepted to the *Proceedings of the 11th International Conference on Artificial Intelligence in Education*. Sydney, Australia, pp. 157-164.

DeBello, T.C. (1990).Comparison of eleven major learning styles models: variables, appropriate populations, validity of instrumentation and the research behind them. *Journal or Reading, Writing, and Learning Disabilities*, 6, pp. 203-222.

De Bra, P. (1998). Adaptive Hypermedia on the Web: Methods, Technology and Applications. In *The Proceedings of Webnet*, Toronto, Canada.

De Bra P., Aerts A., Smits D., and Stash N. (2002) AHA! Version 2.0: More Adaptation Flexibility for Authors. In *Proceedings of World Conference on E-Learning*. Driscoll M. and Reeves T. C. (Eds.) 15-19, 2002, AACE, Montreal, Canada, pp. 240-246.

De Bra P., Brusilovsky P. and Houben G. (1999), 'Adaptive Hypermedia: From Systems for Framework', *ACM Computing Surveys*, Vol. 31, Number 4es.

De Bra P. and Calvi L. (1998). AHA! An open Adaptive hypermedia Architecture. *The New Review of Hypermedia and Multimedia 4*, pp. 67-88.

De Rosis F., De Carolis B., Pizzutilo S. (1994). User Tailored Hypermedia Explanations. In *The Workshop on Adaptive Hypertext and Hypermedia at 4th International Conference on User Modeling*, MA, USA.

Draper N. R., and Smith H., (1981). Applied Regression Analysis, 2nd ed. Wiley, New York.

Fink J., Kobsa A., and Schreck J (1997). Personalized hypermedia information provision through adaptive and adaptable systems features: User modeling, privacy and security issues. In *Intelligence in Services and Networks: Technology for Cooperative Competition*, A. Mullery, M. Besson, M. Campolargo, R. Gobbi, and R. Reed, (Eds.) *Springer-Verlag*, pp. 459-467.

Fischer S., and Steinmetz R. (2000). Automatic Creation of Exercises in Adaptive Hypermedia Learning Systems. In *The Proceedings of ACM Hypertext,* San Antonio, TX, USA. pp. 49-55.

Fox A. (1982). Quiz time in the classroom: Spelling Bee. *Creative Computing*, 8(10), pp. 18-20.

Carro R. Pulido E., Rodriguez P.(1999). TANGOW, Task-Based Adaptive Learner Guidance on the WWW. In *The Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.*

Gates K. Lawhead P. Wilkins D. (1998). A Design for Delivering Filtered Web Views. In *The Proceedings of Webnet*, Toronto, Canada.

Geldof S. (1998). Con-textual Navigation Support. *The New Review of Hypermedia and Multimedia 4*, pp. 47-65.

Gertner, A., Conati, C., and VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. *Proceedings of the 15th National Conference on Artificial Intelligence,* Cambridge, MA: The MIT Press. pp. 106-111.

Goecks J., Shavlik J. (2000). Learning Users' Interests by Unobtrusivly Observing Their Normal Behavior. In *International Conference on Intelligent User Interfaces.*

Gonschorek M., Herzog C. (1995). Using Hypertext for an Adaptive Help system in an Intelligent Tutoring System. In *AI-ED'95, 7th World Conference on Artificial Intelligent in Education*, Washington DC, pp. 274-281.

Hartman V. F. (1995). Teaching and learning style preferences: Transitions through technology. *VCCA Journal 9*, no. 2, pp. 18-20.

Henze N. and Nejdl W. (1999). Adaptivity in the KBS Hyperbook System. In *The Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.*

Henze, N., Nejdl, W., and Wolpers, M. (1999). Modeling constructivist teaching functionality and structure in the KBS hyperbook system. *In CSCL'99: Computer Supported Collaborative Learning*, Standford, USA.

Henze N. and Nejdl W. (2000). Extendible Adaptive Hypermedia Courseware, Integrating Different Courses and Web Material. In *The Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, Italy, pp. 109-120.

Hill, J. E. (1981). The educational sciences: A conceptual framework. West Bloomfield Hills, MI: Hill Educational Sciences Foundation.

Hoffman R. (1987). The Problem of Extracting the Knowledge of Experts From the Perspective of Experimental Psychology. *AI Magazine*, pp. 53-67.

Höök K., Karlgren J., Waern A., Dahlbäck N., Jansson C., Karlgren K., and Lemaire B(1996). A glass box approach to adaptive hypermedia. *User Modeling and User Adapted Interaction 6*, 2-3 pp. 157-184.

Ikeda M. & Mizoguchi R. (1994). FITS: A Framework for ITS--A computational model of tutoring. J. of Artificial Intelligence in Education 5(3), pp. 319-348.

Kaplan C. Fenwick J. and Chen J. (1998). Adaptive Hypertext Navigation Based on User Goals and Context. In *Adaptive Hypertext and Hypermedia*. Brusilovsky P. Kobsa A. and Vassileva J. (Eds.), Kluwer Academic Publishers, The Netherlands, ch.2, pp. 45-70.

Kay J., and Kummerfeld R (1997). User Models for Customized Hypertext. In *Intelligent hypertext: Advanced Techniques for the World Wide Web,* C. Nicholas and J. Mayfield, (Eds.), Springer, LNCS Vol. 1326.

Kay J., and Kummerfeld R (1994). An Individualized Course for the C Programming Language. In *Second International WWW Conference*, Chicago, IL.

Kettel L., Thomson J., and Greer J. (2000). Generating Individualized Hypermedia Applications. In *The International Workshop on Adaptive and Intelligent Web-based Educational Systems, held in conjunction with ITS 2000*, Montreal, Canada.

Kobsa A., Nill A., and Fink J. (1997). Adaptive Hypertext and hypermedia Clients of the User Modeling System BGP-MS. In *Intelligent Multimedia Information Retrieval*, Maybury M. (Ed.), MIT Press, Boston.

Koedinger, K.R., Corbett, A. & Anderson, J.R. (1997). Intelligent tutoring systems. In *Handbook of human- computer interaction 2nd Edition*. Helander M., Landauer T. K, Prabhu P. (Eds.) New York, Elsevier.

Kolb, D. A. (1981). Experiential learning theory and the learning style inventory: A reply to Friedman and Stumpf. *Academy of Management Review,* 6(2), pp. 289-296.

Koschmann T. (1996). Paradigm shifts and instructional technology: An introduction. In *CSCL: Theory and practice.* Koschmann T. (Ed.) Mahwah, NJ: Lawrence Erlbaum Associates.

Kuhn T.S. (1972). *The structure of scientific revolutions* (2nd ed.). Chicago, IL: Chicago University Press.

Kushmerick N., McKee J., and Toolan F. (2000). Towards Zero-Input Personalization, Referrer-Based Page Prediction. In *The Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, Italy, pp. 133-143.

Kushniruk A., Wang H. (1994). A Hypermedia-Based Educational System with Knowledge-Based Guidance. In *ED-MEDIA'94 – World Conference on Educational Multimedia and Hypermedia*, Vancouver, Canada, pp. 335-340.

Jameson A. (1999). What can the rest of us learn from research on adaptive hypermedia and vice versa? Tech. rep. University of Saarbrücken.

Joerding T. (1999). A Temporary User Modeling Approach for Adaptive Shopping on the Web. In *The Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.*

Judd C.M., Kenny D.A., & McClelland G.H. (2001). Estimating and testing mediation and moderation in within-participant designs. *Psychological Methods*, in press.

Lajoie S.P. and Vivet M. (1999). Open Learning Environments: New Computational Technologies to Support Learning, Exploration and Collaboration, *Artificial Intelligence in Education, IOS Press,* Amsterdam.

Langenbach C. and Bodendorf F. (1997). A Framework for WWW-based Learning with Flexible Navigational Guidance. In *The Proceedings of Webnet*, pp. 189-201.

Lepper M., Woolverton M., Mumme D., & Gurtner J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In *Computers as cognitive tools,* Lajoie S. & Derry S. (Eds.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lieberman H. Letizia (1995). An Agent That Assist Web Browsing. In *The Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. Chris S. Mellish (Eds.), Morgan Maufmann publisher Inc. San Mateo, CA, USA pp. 924-929.

Lin F., Danielson R., and Herrgott S. (1996). Adaptive Interaction through WWW. In *Educational Telecommunications*, pp. 173-178.

Litzinger M.E., and Bonnie O. (1993). Accommodating diverse learning styles: Designing instruction for electronic information sources. In What is Good Instruction Now? *Library Instruction for the 90s*. Shirato L.. Arbor A. (Eds.), MI: Pierian Press.

McCann P.H. (1975). Training mathematics skills with games. *Navy Personnel Research and Development Center*: San Diego, CA.

Melis E., Ullrich C. (2003). How to teach it Polya-Inspired Scenarios in ActiveMath. In *AI-ED, Shaping the future of learning through Intelligent Technologies.* Ulrich Hoppe, Felisa Verdejo, Judy Kay (Eds). pp 141-147.

Micarelli A. Sciarrone F. (1996). A Case-Based Toolbox for Guided Hypermedia Navigation. In *Fifth International Conference on User Modeling* Hawaii, pp. 129-136.

Murray T. Shen T. Piemonte J. Condit C. and Thibedeau J. (2000). Adaptivity for Conceptual and Narrative Flow in Hyperbooks: The Metalinks System. In *The Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, Italy, pp. 155-166.

Nakatani L.H., Egan D.E., Ruedisueli L.W., Hawley P.M., & Lewart D.K. (1986). TNT: A talking tutor 'n' trainer for teaching use of interactive computer systems. *CHI '86 Proceedings*.

Neiman D., Eliot C., and Lamar M. (1998) Medtec: A Web-Based Intelligent Tutoring for Basic Anatomy. In *The Proceedings of Webnet,* Toronto, Canada, pp. 208-212.

Pérez T., Gutiérrez J., Lopistéguy P. (1995). The Role of Exercises in a User-Adapted Hypemedia. In *3rd International Conference on Computer Aided Engineering Education, CAEE '95,* Bratislava, Slovakia.

Pérez T., Gutiérrez J., Lopistéguy P. (1995). An Adaptive Hypermedia System. In *AI-ED'95, 7th World Conference on Artificial Intelligent in Education*, USA, pp. 351-358.

Pilar de Silva D., Van Durm R., Duval E., and Olivie H. (1998). Concepts and Documents for Adaptive Educational Hypermedia, a Model and a Prototype. In *The 2nd Workshop on Adaptive Hypertext and Hypermedia, HYPERTEXT'98*, Pittsburg, USA.

Prentzas J, Hatzilygeroudis I, and Garofalakis J. (2002). A Web-Based Intelligent Tutoring System Using Hybrid Rules as Its Representational Basis. Intelligent Tutoring Systems, pp. 119-128.

Pressey S.L. (1927). A machine for automatic teaching of drill material. *School and Society*, 25, pp. 549-552.

Pylyshyn Z. (1989). Computing in cognitive science. In *Foundations of cognitive science.* M. Posner (Ed.), Cambridge, MA: MIT Press.

Rabinowitz J., Mathe N. and Chen J. (1995). Adaptive HyperMan, A Customizable Hypertext System for Reference Manuals. In *The Proceedings of the Workshop on AI Applications in Knowledge Navigation and Retrieval*, Cambridge, MA.

Rivlin E., Botafogo R. Schneidermann B. (1994). Navigating in Hyperspace, Designing a Structure-Based Toolbox. *Communications of the ACM 37*, pp. 87-96.

Roberts, J., Pane, J., Stehlik, M., and Carrasquel, J. (1988). "The Design View: A Design Oriented High Level Visual Programming Environment." *Proceedings of 1988 IEEE Workshop on Visual Language*, Pittsburgh, PA.

Rousseau F., Garcia J., Valdeni de Lima J., and Duda A. (1999). User Adaptive Multimedia Presentations for the WWW. In *The Eight International World Wide Web Conference*, Toronto, Canada.

Salchian B., Aïmeur E. (2003). A Multimedia Training System Applied to Telephony. *Information Technology Based Higher Education and Training ITHET03.*

Saiz F., Szekely P., Devang P. (1998). Customized Web-Based Data Presentation. In *The Proceedings of Webnet*, Toronto, Canada.

Schreiner M.E. (2001). The Role of Automatic Feedback in the Summarization of Narrative Text. Unpublished doctoral dissertation, University of Colorado, Boulder.

Shastri L. (1991). Why Semantic Network? *Principles of Semantic Networks: Explorations in the Representation of Knowledge,* John F. Sowa(Ed.), pp 109-136.

Shute V.J. and Regian J.W. (1999). Rose Garden Promises of Intelligent Tutoring Systems: Blossom or Thorn? *Space Operations, Automation and Robotics Conference*, Albuquerque, NM.

Shute V, (1995). Smart evaluation: Cognitive diagnosis, mastery learning and mediation. In *The Proceedings of Artificial Intelligent in Education*, pp. 123-130.

Specht M. and Opperman R. (1998). ACE – Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia 4* (1998), 141 – 162.

Stefani A. Strapparava C. (1998). Personalizing Access to Web Sites, the SiteIF Project. In *2nd Workshop on Adaptive Hypertext and Hypermedia*, *HYPERTEXT'98*, Pitsburg, USA, pp. 20-24.

Vassileva J. (1998). A Task-Centered Approach for User Modeling in a Hypermedia Office Documentation System. In *Adaptive Hypertext and Hypermedia,* Brusilovsky P., Kobsa A., and Vassileva J. (Eds.), Kluwer Academic Publishers, Netherlands, pp. 209-247.

Weber G. and Specht, M. (1997). User Modeling and Adaptive Navigation Support in WWW-based tutoring systems. In *Proceedings of the Sixth International Conference on User bModeling*, *UM97,* Sprinter Wien New York.

Woolf B. P. (1992). AI in Education. *Encyclopedia of Artificial Intelligence*, Shapiro, S., (Ed.), John Wiley & Sons, Inc., New York, pp. 434-444.

Woolf B. P., (2002). *Building intelligent tutors,* Department of Computer Science University of Massachusetts Amherst, MA 01003.

Woolf B. P., & McDonald D.D. (1984). Building a computer tutor, Design issues. *IEEE Computer*, 17, pp. 61-73.

Zyryanov M. (1996). Adaptive Local Maps in the Hyper Components of Intelligent Learning Environments. In *Multimedia, Hypermedia and Virtual Reality, Models, Systems, and Applications*. Brusilovsky P., Kommers, P. and Streitz N. (Eds.), Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp.306-310.

Ziemer Stephan, http://www.sts.tu-harburg.de/~st.ziemer/its.pdf