INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Providing Adaptation for an Adaptive Hypermedia System

MARTA MAFALDA DE ALMEIDA FERNANDES

Outubro de 2015

POLITÉCNICO
DO PORTO

Providing Adaptation for an Adaptive Hypermedia System

Marta Mafalda de Almeida Fernandes

Dissertation to obtain the Master of Science degree in Computer Science, Specialization in Knowledge-based and Decision Support Technologies

Supervisor: Luiz Faria

Co-Supervisor: Constantino Martins

Jury:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, October 2015

Resumo

O ensino à distância cresceu consideravelmente nos últimos anos e a tendência é para que continue a crescer em anos vindouros. No entanto, enquanto que a maioria das plataformas de ensino à distância utilizam a mesma abordagem de ensino para todos os utilizadores, os estudantes que as usam são na realidade pessoas de diferentes culturas, locais, idades e géneros, e que possuem diferentes níveis de educação.

Ao contrário do ensino à distância tradicional, os sistemas de hipermédia adaptativa educacional adaptam interface, apresentação de conteúdos e navegação, entre outros, às características, necessidades e interesses específicos de diferentes utilizadores.

Apesar da investigação na área de sistemas de hipermédia adaptativa já estar bastante desenvolvida, é necessário efetuar mais desenvolvimento e experimentação de modo a determinar quais são os aspetos mais eficazes destes sistemas e avaliar o seu sucesso. A Plataforma de Aprendizagem Colaborativa da Matemática (PCMAT) é um sistema de hipermédia adaptativa educacional com uma abordagem construtivista, que foi desenvolvido com o objetivo de contribuir para a investigação na área de sistemas de hipermédia adaptativa. A plataforma avalia o conhecimento do utilizador e apresenta conteúdos e atividades adaptadas às características e estilo de aprendizagem dominante de estudantes de matemática do segundo ciclo.

O desenvolvimento do PCMAT tem também o propósito de auxiliar os alunos Portugueses com a aprendizagem da matemática. De acordo com o estudo PISA 2012 da OCDE [OECD, 2014], o desempenho dos alunos Portugueses na área da matemática melhorou em relação à edição anterior do estudo, mas os resultados obtidos permanecem abaixo da média da OCDE. Por este motivo, uma das finalidades deste projeto é desenvolver um sistema de hipermédia adaptativa que, ao adequar o ensino da matemática às necessidades específicas de cada aluno, os assista com a aquisição de conhecimento.

A adaptação é efetuada pelo sistema usando a informação constante no modelo do utilizador para definir um grafo de conceitos do domínio específico. Este grafo é adaptado do modelo do domínio e utilizado para dar resposta às necessidades particulares de cada aluno. Embora a trajetória inicial seja definida pelo professor, o percurso percorrido no grafo por cada aluno é determinado pela sua interação com o sistema, usando para o efeito a representação do conhecimento do aluno e outras características disponíveis no modelo do utilizador, assim como avaliação progressiva.

A adaptação é conseguida através de alterações na apresentação de conteúdos e na estrutura e anotações das hiperligações. A apresentação de conteúdos é alterada mostrando ou ocultando cada um dos vários fragmentos que compõe as páginas dum curso. Estes fragmentos são compostos por diferentes objetos de aprendizagem, tais como exercícios,

figuras, diagramas, etc. As mudanças efetuadas na estrutura e anotações das hiperligações têm o objetivo de guiar o estudante, apontando-o na direção do conhecimento mais relevante e mantendo-o afastado de informação inadequada.

A escolha de objectos de aprendizagem adequados às características particulares de cada aluno é um aspecto essencial do modelo de adaptação do PCMAT. A plataforma inclui para esse propósito um módulo responsável pela recomendação de objectos de aprendizagem, e um módulo para a pesquisa e recuperação dos mesmos. O módulo de recomendação utiliza lógica Fuzzy para converter determinados atributos do aluno num conjunto de parâmetros que caracterizam o objecto de aprendizagem que idealmente deveria ser apresentado ao aluno. Uma vez que o objecto "ideal" poderá não existir no repositório de objectos de aprendizagem do sistema, esses parâmetros são utilizados pelo módulo de pesquisa e recuperação para procurar e devolver ao módulo de recomendação uma lista com os objectos que mais se assemelham ao objecto "ideal". A pesquisa é feita numa árvore k-d usando o algoritmo k-vizinhos mais próximos. O modelo de recomendação utiliza a lista devolvida pelo módulo de pesquisa e recuperação para seleccionar o objecto de aprendizagem mais apropriado para o aluno e processa-o para inclusão numa das páginas Web do curso.

O presente documento descreve o trabalho desenvolvido no âmbito do projeto PCMAT (PTDS/CED/108339/2008), dando relevância à adaptação de conteúdos.

Palavras-chave: Hipermédia Adaptativa Educacional, Modelo do Utilizador, Modelo de Adaptação, Objectos de Aprendizagem, Constructivismo, Estilos de Aprendizagem

Abstract

E-learning has grown considerably in recent years, and current trends point towards continuous growth in the coming years. However, while most e-learning platforms offer a one-size-fits-all solution, students are actually people of different cultures, locations, ages, genders, and with varying levels of education.

Unlike traditional e-learning approaches, adaptive educational hypermedia systems adapt interface, content presentation, and link navigation, among others, to the specific characteristics, needs and interests of different users. As these goals and characteristics change, so does the content presented by the system.

The Mathematics Collaborative Learning Platform (PCMAT) is an adaptive educational hypermedia system with a constructivist approach, which assesses the user's knowledge and presents contents and activities adapted to the characteristics and dominant learning style of students of mathematics in basic schools.

PCMAT contains a user model, where information about the user is stored; a domain model, which consists of a semantic network of domain concepts; and an adaptation model that defines the adaptation rules and the interaction mechanisms between the user and the system. By relating the information contained in the user model to the domain model the system can, through the adaptation model, adapt its content, navigation and interface to each user's specific needs.

This document presents the work performed within the scope of the PCMAT project (PTDS/CED/108339/2008), with a focus on content adaptation. This project aims to contribute to the research on adaptive hypermedia systems, as well as demonstrate how such systems can improve learning in a basic school environment.

Keywords: Adaptive Educational Hypermedia, User Model, Adaptation Model, Learning Objects, Constructivism, Learning Styles

Acknowledgements

I would like to begin by thanking my supervisors, Professor Luiz Faria and Professor Constantino Martins, for all their work and support during the development of PCMAT (PTDS/CED/108339/2008), for giving me the opportunity to work on this project, and for creating a very pleasant and motivating working environment.

I would also like to thank my colleague Paulo Couto for his work and helpful advice.

In addition, I would like to thank Daniel Mota for his encouragement, and Virgínia Nascimento for her suggestions.

Finally, I would like to thank GECAD for providing the resources and conditions necessary to work on PCMAT.

Table of Contents

1 Introduction 1

1.1 Main Goals 3

1.2 Main Contributions 3

1.3 Methodological Approach 4

1.4 Document Structure 6

2 Background 7

2.1 Learning Theories 7
- 2.1.1 Behaviorism 7
- 2.1.2 Cognitivism 11
- 2.1.3 Constructivism 12

2.2 Learning Styles 13
- 2.2.1 David Kolb's Experiential Learning Model 14
- 2.2.2 VAK Model 17

2.3 Adaptive Hypermedia Systems 18
- 2.3.1 The User/Student Model 20
- 2.3.2 The Domain Model 22
- 2.3.3 The Adaptation Model 23

2.4 Adaptive Hypermedia Architecture 29
- 2.4.1 Architecture 29
- 2.4.2 The Domain/Adaptation Model and the User Model 30
- 2.4.3 Adaptation 32
- 2.4.4 Authoring Tools 39

3 Mathematics Collaborative Learning Platform - PCMAT 43

3.1 Introduction 43

3.2 Architecture 43

3.3 Student Model Implementation 45

3.4 Domain Model Implementation 49

3.5 Learning Objects 52
- 3.5.1 A metadata standard for PCMAT 52
- 3.5.2 IEEE Learning Object Metadata 53
- 3.5.3 Authoring Tool for the creation of Assessment Learning Objects 54

4 Adaptation in PCMAT 59

4.1 Adaptation Model 59
- 4.1.1 Adaptation Rules 60
- 4.1.2 Adaptation Mechanisms 61
- 4.1.3 Module for the assessment of open-ended questions 65

4.2 Learning Objects Recommendation 66

4.2.1 Recommendation Module 66
4.2.2 Search and Retrieval Module 70

5 Evaluation Results 73

5.1 Description of the subjects involved in the experiment 73

5.2 Study Design 74

5.3 Final Test Results 75

5.4 Learning Preferences 77

5.5 Results of the assessment questionnaires 77
5.5.1 Acceptance and usability 78
5.5.2 Usefulness and difficulty using PCMAT's functionalities 80
5.5.3 Frequency of usage 81

6 Conclusion and Future Work 83

6.1 Conclusion 83

6.2 Limitations and Future Work 84

References 87

List of Figures

Figure 1 – The action research process [dos Santos and Travassos, 2011]. 5
Figure 2 – Results from the study of latent learning in rats [Tolman and Honzik, 1930]. 10
Figure 3 – David Kolb's Experiential Learning Model [Chapman, 2009]. 15
Figure 4 – The student model. 21
Figure 5 – Adaptation techniques in adaptive hypermedia [Brusilovsky, 1996b]. 25
Figure 6 – AHA!'s architecture [De Bra et al, 2003]. 30
Figure 7 – Definition of the resources associated with an object.. 35
Figure 8 – Association between a resource and a link [De Bra et al, 2006]............................. 39
Figure 9 – The Graph Author [De Bra et al, 2006].. 41
Figure 10 - Detailed representation of PCMAT's architecture... 45
Figure 11 - Questionnaire used to determine users' learning styles. 46
Figure 12 - PCMAT's Domain dependent data architecture [Martins et al, 2013]. 48
Figure 13 – The graphs GPCMAT (black edges) and GI (red edges). 51
Figure 14 – Interface for the creation of activities. 55
Figure 15 – Interface for the creation of parameterized activities. 56
Figure 16 – Evaluation of the adaptation rules associated with the access attribute. 62
Figure 17 - Communication between modules. 69

List of Tables

Table 1 – Kolb's Learning Styles Inventory. 16
Table 2 - Characteristics used in the student model [Martins et al, 2013]. 47
Table 3 - Mapping of learning styles. 48
Table 4 – Learning preference by group. 74
Table 5 - Results of the Kolmogorov-Smirnov test applied to each concept. 75
Table 6 - Statistical comparison for each concept (Mann-Whitney test). 76
Table 7 – Changes in learning preferences. 77
Table 8 - Scale used to assess PCMAT's acceptance and usability 78
Table 9 - Results concerning the acceptance and usability of PCMAT. 79
Table 10 - Scale used to assess the usefulness of PCMAT's functionalities. 80
Table 11 - Scale used to assess the difficulty in using PCMAT's functionalities. 80
Table 12 - Results concerning the usefulness of PCMAT's functionalities. 81
Table 13 - Results concerning the difficulty of use of PCMAT's functionalities. 81
Table 14 – Frequency of usage of PCMAT. 82

Acronyms and Symbols

List of Acronyms

AEHS	Adaptive Educational Hypermedia System
AHA!	Adaptive Hypermedia Architecture
AHAM	Adaptive Hypermedia Application Model
AHS	Adaptive Hypermedia System
AM	Adaptation Model
DCMES	Dublin Core Metadata Element Set
DM	Domain Model
FCL	Fuzzy Control Language
FCT	Portuguese Foundation for Science and Technology
GECAD	Knowledge Engineering and Decision Support Research Group
IEEE	Institute of Electrical and Electronics Engineers
LO	Learning Objects
LOM	Learning Object Metadata
MOOC	Massive Online Open Course
NISO	National Information Standards Organization
PCMAT	Mathematics Collaborative Learning Platform
SM	Student Model
URI	Uniform Resource Identifier
VAK	Visual, Auditory, Kinesthetic

List of Symbols

σ	Standard Deviation
p	P-value

1 Introduction

E-learning has been gaining prominence over the past decade, but it has seen some significant changes since 2012 due to the rapid spread of massive online open courses (MOOCs) [Masters, 2011.

MOOCs are free online courses aimed at large scale participation. They have existed for a few years [Masters, 2011], but after more than a hundred and sixty thousand students in over a hundred and ninety countries enrolled in Sebastian Thrun and Peter Norvig's "Introduction to Artificial Intelligence" in September 2011 [Udacity, 2011], several free online learning platforms have launched that now offer courses on various subjects. Currently, approximately fifteen million students have enrolled in the courses of Coursera, a social entrepreneurship company that partners with top US universities and was founded in January 2012 [edSurge, 2015] [Coursera, 2015]. Udacity, founded by Sebastian Thrun and two colleagues after the success of "Introduction to Artificial Intelligence", now serves about four million students [edSurge, 2015][Udacity, 2011]. And edX, a nonprofit start-up from Harvard University and the Massachusetts Institute of Technology, already hit the five million student mark [edSurge, 2015][edX, 2015].

MOOCs are not massive in the number of students alone; there is also great diversity in the people enrolling. Students of these courses include both men and women from all over the world, with varying levels of education and ranging from preadolescents to senior citizens. As can be expected, these students do not all learn in the same way or with the same ease, yet MOOCs, as is the case with most e-learning, offer a one-size-fits-all solution.

Unlike traditional e-learning approaches, Adaptive Educational Hypermedia Systems (AEHSs) adapt interface, content presentation, and link navigation, among others, to the specific characteristics, needs and interests of different users [Brusilovsky et al, 2007] [De Bra, 2006]. The aim of these systems is to help users achieve their learning goals, therefore characteristics such as previous knowledge and learning styles are particularly important [De Bra, 2006]. Adaptive Hypermedia Systems (AHS) usually consist of three interdependent modules: user model, domain model and adaptation model [Benyon, 1993] [De Bra et al, 2004]. By relating the user model to the domain model the system can, through the adaptation model, adapt its content, navigation and interface to each user's specific needs.

AEHS offer an educational experience that is tailored to each individual student and as e-learning continues to evolve and grow, they are a solution to a problem that is particularly noticeable in large-scale e-learning projects such as MOOCs: the absence of a teacher that will guide students and provide them with individual explanations, adapted to their specific abilities, knowledge and personality. AEHS can adapt content presentation to suit each student's different level of knowledge and in that way improve their learning experience.

AHS have been the subject of much research but more development, experimentation and implementation are necessary to conclude about the adequate features and effectiveness of these systems [Martins et al, 2008a].

Learning styles represent models of how a person learns. Recent studies have shown the majority of people are multimodal, meaning they have more than one learning style [Fleming, 2007] [Miller, 2001]. While academia has yet to reach an agreement on whether learning styles are effective or not, there doesn't seem to be any evidence suggesting their use is detrimental. The mathematics teachers working on this project have been teaching for many years and their personal opinion, based on what they have observed, is that learning styles might indeed be useful. For that reason, and in an attempt to help determine whether learning styles are useful or not, we have decided to apply this theory when developing the Mathematics Collaborative Learning Platform (PCMAT).

The PCMAT platform [Martins et al, 2011] is an AEHS possessing innovative features that was developed with the aim of helping drive AHS research forward. PCMAT has a constructivist approach, and assesses the user's knowledge and presents contents and activities adapted to the characteristics and learning preferences of students of mathematics in basic schools.

This project also serves the purpose of assisting Portuguese students improve their knowledge of mathematics. According to the OECD PISA 2012 study [OECD, 2014], Portugal has made progress in mathematics performance, with most improvements occurring between 2006 and 2009. However, in spite of these good results Portugal is still below the OECD average in mathematics performance. Portugal's lower rank is 36 and upper rank is 26 out of 65 countries featured in the study. There's clearly room for improvement and Portugal still has a long way to go before we can consider ourselves satisfied. With this project we hope to develop an adaptive hypermedia system that'll help improve these results by tailoring the way in which basic school mathematics is taught to each student's individual needs.

1.1 Main Goals

The PCMAT project (PTDS/CED/108339/2008) was developed with the purpose of contributing to the research in the field of adaptive hypermedia, as well as helping students of mathematics improve their knowledge. As such, PCMAT's goals include:

- The Development of an AEHS to support and improve the learning of mathematics in the context of basic schools.
- The definition of the student model attributes.
- The production of learning objects aligned with an adopted standard.

More specifically, the author of the present thesis has the following goals:

- The development of PCMAT's adaptation model.
- The integration of the adaptation model with the PCMAT platform.

1.2 Main Contributions

This document presents the research and development performed within the scope of the PCMAT project, with a focus on content adaptation. This project was undertaken at the Knowledge Engineering and Decision Support Research Group (GECAD) and funded by the Portuguese Foundation for Science and Technology (FCT). The results obtained with the PCMAT platform demonstrate the usefulness of AEHS, and show how the Mathematics knowledge of basic school students can be improved by adapting content and activities to each student's specific goals, characteristics and learning styles.

This project's contributions to the advancement of Adaptive Educational Hypermedia Systems include:

- The definition of a new architecture and strategies for the implementation of an Adaptive Hypermedia Educational platform to support and improve Mathematics in basic schools.
- Adaptation based on constructivism and learning styles.
- The conception and development of standard-compliant learning objects.
- The definition of a Student Model that supports adaptive functionalities based on learning styles, constructivism and standard-compliant learning objects.
- The implementation of a set of adaptive and dynamic educational strategies.

In particular, the author of this thesis contributed with the following:

- The development and implementation of PCMAT's adaptation model.
- The integration of the adaptation model into the preexisting PCMAT system and AHA!'s adaptation engine.
- The development of a module that uses Fuzzy Logic to choose learning objects that are adapted to a user's knowledge and characteristics.
- The development of an authoring tool for the creation of assessment learning objects.

- The implementation of a k-d tree as the data structure where the URI of the system's learning objects are stored.
- The development of a module for the evaluation of open-ended questions.

Furthermore, the advances attained with the PCMAT project led to the publication of the following scientific papers:

- Fernandes M, Couto P, Martins C, Faria L, Content Adaptation for an Adaptive Hypermedia System, Proceedings of INTELLI 2013, The Second International Conference on Intelligent Systems and Applications, Venice, Italy, 21-26 April 2013, pp 54-57.
- Fernandes M, Couto P, Martins C, Faria L, Bastos C, Costa F, Learning Objects Recommendation in an Adaptive Educational Hypermedia System, Proceedings of the 8th WSEAS International Conference on Educational Technologies (EDUTE '12), Included in ISI/SCI Web of Science and Web of Knowledge, Porto, Portugal, 2012, pp. 123-128.
- Fernandes M, Martins C, Faria L, Couto P, Valente C, Bastos C, Costa F, Carrapatoso E, Adaptation Model for PCMAT – Mathematics Collaborative Learning Platform, Highlights on Practical Applications of Agents and Multi-Agent Systems, AISC 156, Springer-Verlag, Berlin Heidelberg, 2012, pp. 95-10.
- Couto, P., Fernandes, M., Martins, C., Faria, L. PCMAT, an Adaptive Learning Platform. In Kae Dal Kwack, Franz Ko (Eds.), ICIPT 2013: 8th International Conference on Information Processing, Management and Intelligent Information Technology (ICIPM, ICIIP), pp. 167-170. Seoul: AICIT, 2013. ISBN: 978-1-4503-1783-2.
- Martins C, Faria L, Fernandes M, Couto P, Bastos C, Carrapatoso E, PCMAT Mathematics Collaborative Educational System, Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends, Peña Ayala, Alejandro (Ed.), Vol. 17, Springer, Berlin Heidelberg, 2013, pp. 183-212.
- Martins C, Couto P, Fernandes M, Bastos C, Lobo C, Faria L, Carrapatoso E, PCMAT - Mathematics Collaborative Learning Platform, Highlights in Practical Applications of Agents and Multiagent Systems, AISC 89, Springer-Verlag, Berlin Heidelberg, 2011, pp. 93-100.
- Couto P, Martins C, Faria L, Fernandes M, Carrapatoso E, PCMAT Metadata Authoring Tool, ISCIES 2011, International Symposium on Computational Intelligence for Engineering Systems, Coimbra, 2011.

1.3 Methodological Approach

The work described in this document was guided by the Action Research methodology. Action Research is focused on solving real-world problems. It essentially consists in identifying a problem, designing and developing a solution to solve it, testing and evaluating the solution, and starting again if the results are not satisfactory [dos Santos and Travassos, 2011][Easterbrook et al, 2008].

Action Research involves both trying to solve a real issue and studying the process by which that solution is attained. This implies collaborative work on the part of all participants, whether researchers or clients (problem owners). In this sense, everyone involved in the project is a co-researcher motivated to solve the problem by active participation and mutual learning. It is often the case that the initiating researcher is himself the client [dos Santos and Travassos, 2011][Easterbrook et al, 2008].

Gerald Susman [Susman and Evered, 1978] devised a process for Action Research that is composed of five phases (Figure 1).

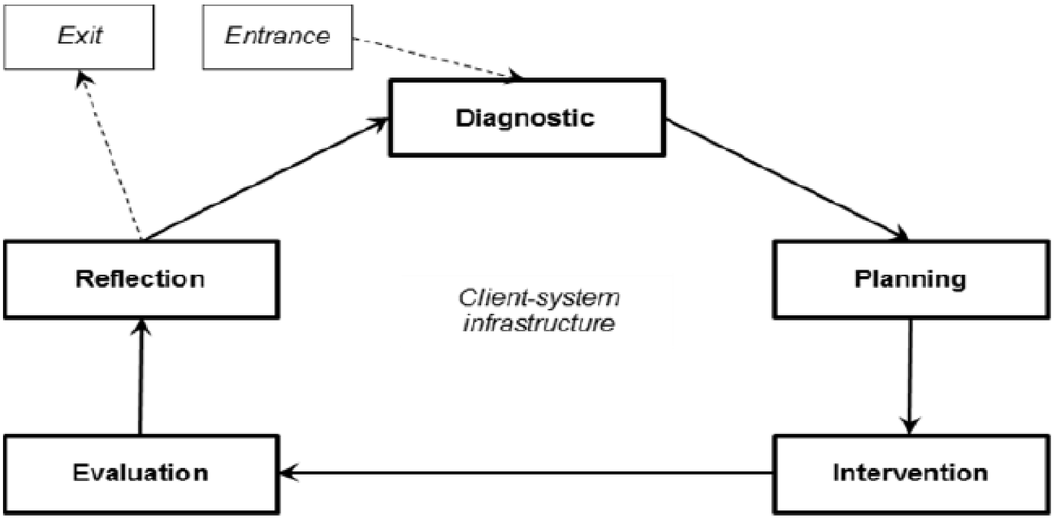

Figure 1 – The action research process [dos Santos and Travassos, 2011].

In the diagnostic phase, the problem is identified and information is gathered about its nature. The phase that follows, planning, implies considering possible solutions and choosing the one that will be implemented. The intervention phase is the one during which the solution previously adopted is put into practice. This is followed by the evaluation phase that involves analyzing the results obtained from the previous phase in regards to the theoretical background. Finally, the reflection phase involves sharing the acquired knowledge among participants and reassessing the problem. The study will end if everyone involved is satisfied with the outcome, otherwise the process can be repeated until an adequate result is achieved [dos Santos and Travassos, 2011].

The Action Research method was applied to this work due to the project's main goal of improving the mathematics knowledge of Portuguese students, thus addressing a real-world problem. The participants involved in the project include researchers, some of whom are educators themselves, and teachers of mathematics in basic schools. The proposed solution is the use by students, both in school and at home, of an adaptive educational system as a learning aid. This study resulted in the development of a prototype that is rooted in educational theories and adaptive hypermedia research. The prototype was tested in two basic schools and the results obtained were analyzed, allowing the participants to conclude about the positive outcomes of the study and the issues that need to be improved. The knowledge obtained with this study was shared with the scientific community through the publication of scientific papers.

1.4 Document Structure

This thesis contains six chapters and is organized as follows:

Chapter 1 describes the motivations behind project PCMAT and gives a general introduction of some of the concepts involved. It also describes the methodological approach of the work described and presents the project's main goals and contributions.

Chapter 2 covers the background of the work performed by introducing the relevant technologies and techniques, and presenting important concepts involved in PCMAT's development.

Chapter 3 introduces the Mathematics Collaborative Learning Platform (PCMAT), describing several aspects of the platform such as its architecture, student and domain models and production of learning objects.

Chapter 4 focuses on the adaptation provided by the system, and gives a detailed account of the adaptation model and the selection of suitable learning objects.

Chapter 5 describes the evaluation process and presents the results obtained.

Chapter 6 includes a final assessment of the work performed and addresses future efforts.

2 Background

This chapter presents the techniques and technologies that underlie the work performed. Section 2.1 gives an overview of learning theories and section 2.2 introduces learning styles. Section 2.3 covers adaptive hypermedia systems and, finally, section 2.4 presents the Adaptive Hypermedia Architecture (AHA!).

2.1 Learning Theories

Learning theories are conceptual models of how people learn. They describe how information is apprehended, processed and consolidated. There are several different theories of how people learn, but three important learning theories are Behaviorism, Cognitivism, and Constructivism.

2.1.1 Behaviorism

Behaviorism considers the mind to be a “black box”, disregarding all internal processes such as thoughts and intentions. It focuses on the study of observable and measurable behaviors. Behaviorists believe a person’s behaviors are the result of his response to different environmental stimuli. According to this theory, learning happens when a new behavior is repeated until it becomes automatic. This process is known as conditioning [Good and Brophy, 1990][Graham, 2000].

There are two types of conditioning, classical conditioning and operant conditioning. In classical conditioning the new behavior emerges as a reflex response to stimulus. Its main proponents were American psychologist John Watson and Russian psychologist Ivan Pavlov [Good and Brophy, 1990][Graham, 2000]. With operant conditioning the behavior is reinforced by reward or punishment. If a certain behavior results in a positive consequence, there’s a reinforcement of the behavior, meaning it will likely be repeated, and the consequence is said to be a reward. On the other hand, if the consequence is a negative one it's viewed as a punishment and the frequency with which the behavior occurs decreases. Operant conditioning results in the avoidance of behaviors associated with negative outcomes, and in the repetition of behaviors associated with positive outcomes [Good and Brophy, 1990][Graham, 2000][Fraley, 2001].

The term "behaviorism" was invented by John B. Watson, one of the main influences of the Behaviorist Learning Theory, who believed psychology should be restricted to strictly scientific methods, rejecting focus on introspective states and promoting the theorization of measurable behaviors. Watson's research originally dealt with animals, but he later turned to the study of human behavior. It was his belief that human beings are born with only the emotional reactions of love and rage, as well as a few reflexes. According to Watson, all other behavior is the result of conditioned stimulus-response associations [Good and Brophy, 1990][Graham, 2000].

Watson illustrated the principles of classical conditioning through an experiment with an 11 month old boy and a rat. At first the boy wasn't afraid of the rat, but in the experiment Watson introduced a frightening, loud noise every time the child touched the rat. Because the child was afraid of the noise he eventually learned to associate it with the rat and began avoiding it. The boy was conditioned to fear the rat, a condition that extended to other small animals. Through such experiments Watson demonstrated how conditioning plays a role in emotional responses to certain stimuli, which might explain certain fears and phobias developed by some people [Watson and Rayner, 1920].

The work of Ivan Pavlov with classical conditioning exerted a great influence on Watson's work in particular and behaviorism in general. Pavlov's experiments with dogs are famous. He became interested in experimenting with these animals after seeing them drool even when no food was nearby. Pavlov then realized the animals were fed by people wearing lab coats and that every time they saw someone wearing a lab coat they reacted as if they were going to be fed. To understand how this occurred, Pavlov went about realizing several experiments with dogs, including his most famous experiment which involved a dog, food and a bell [Todes, 2002].

In the beginning of the experiment Pavlov rang the bell in the presence of the dog and observed it didn't cause any reaction in the animal. Pavlov then proceeded to ring the bell every time he fed the dog. After a while, ringing the bell alone was enough to make the dog salivate. The dog learned to associate the sound of the bell with food and what started as an unconditioned response became a conditioned one. Pavlov also noted that the dog salivated not only to the sound of the bell but also to similar sounds and that if he stopped ringing the bell whenever the dog was fed, the animal would eventually cease to salivate when hearing that sound [Todes, 2002].

Operant conditioning was developed by Burrhus Frederic Skinner, chosen as the most influential psychologist of the 20^{th} century in a 2002 survey [Haggbloom et al, 2002]. Like John

Watson he was adamant in his refusal to accept that processes, such as thinking and feeling, had any influence in a person's behavior. In his perspective, behavior occurs as a response to some consequence. In that sense, a positive consequence positively reinforces the behavior that caused it, whereas a negative consequence results in negative reinforcement of the causing behavior. According to Skinner, reinforcement is crucial to shape and control behavior, and both positive and negative reinforcement strengthen behavior. He conceived of the following reinforcement mechanisms: positive reinforcement, negative reinforcement, extinction and punishment. Positive reinforcement consists in rewarding desirable behaviors. For example, if a dog correctly performs a trick he's given a doggy treat. Negative reinforcement happens when a specific behavior helps an individual avoid an undesirable situation, such as brushing one's teeth regularly is likely to keep one out of the dentist's office. Extinction means not reinforcing a behavior. For example, parents are often told the best way to deal with a child throwing a tantrum is ignoring his behavior. Finally, punishment is the negative consequence of a behavior. When a child burns his hands while playing with matches there's a good chance he won't do it again, particular if it happens more than once. Forcing a person or animal to behave in a specific way doesn't always happen immediately. Reaching the desired behavior often means reinforcing behaviors close to the intended one. For example, when teaching a dog to fetch one should reinforce the behavior of bringing the stick back half the way even if the desired outcome is for the dog to bring the stick back all the way. However, the dog shouldn't be rewarded if he doesn't pick up the stick, or takes it elsewhere. In that way, the dog's behavior is shaped until it finally reaches the desired behavior. Once the dog brings the stick back all the way, reinforcement doesn't have to happen every time. Skinner came up with different schedules of reinforcement that produced better results in reinforcing behavior than if reinforcement occurred every time the desired behavior was displayed. Schedules can be fixed or variable and be based on intervals or ratios. With interval schedules reinforcement happens every time a fixed or variable amount of time has passed. Ratio schedules depend on the number (fixed or variable) of times a person or animal displays the desired behavior [Skinner, 1953][Ferster and Skinner, 1957][Good and Brophy, 1990].

Behaviorism soon fell short as a theory that could sufficiently explain learning. Edward Tolman didn't believe behaviorism was all there was to learning. Through experiments performed using rats, Tolman observed latent learning, learning that isn't observable. Tolman and C. H. Honzik [Tolman and Honzik, 1930] had three different groups of rats go through a maze over the course of three weeks. The first group was given cheese upon reaching the end of the maze, thus reinforcing that behavior. The second group of rats was never given any cheese and the third group of rats didn't get any cheese for the first ten days of the experiment, but was rewarded with cheese from the eleventh day onwards. As can be expected, the first group committed far fewer mistakes than the second group of rats whose behavior was never reinforced. However, Tolman observed that, although the third group made a similar amount of mistakes as the first group during the first ten days when it didn't receive any cheese, their results improved dramatically as soon as they began receiving cheese. As can be observed in figure 2, the rats from the third group committed more errors than the second group for the first ten days, but after being given cheese on the eleventh day it took them only two days to catch up to the second group [Tolman and Honzik, 1930].

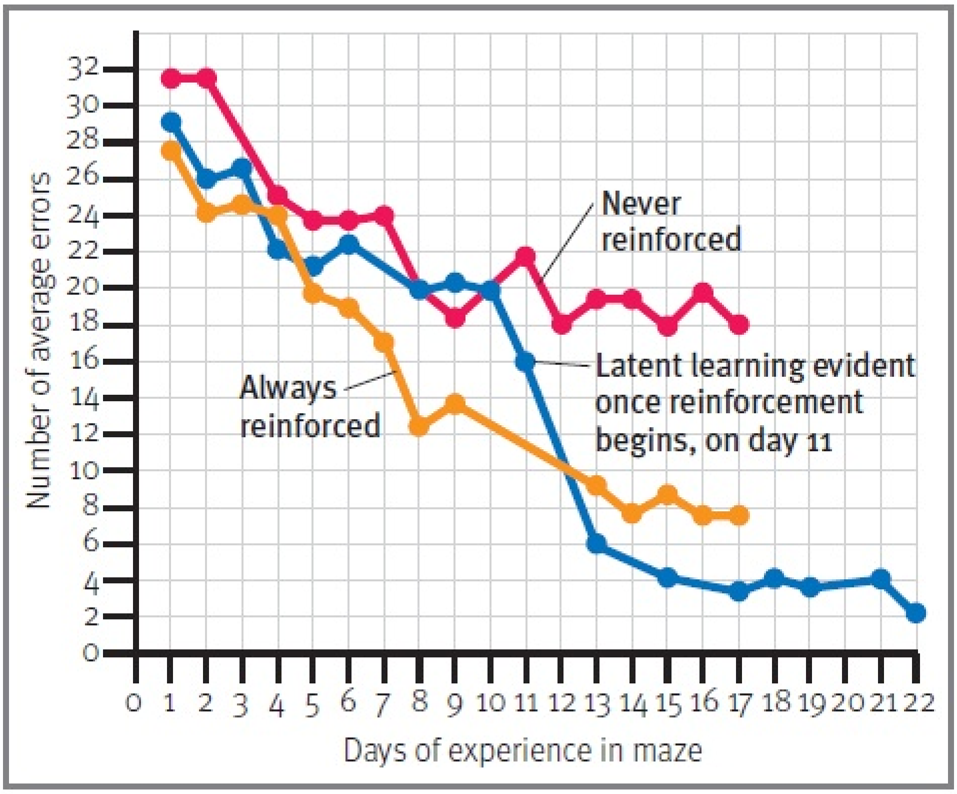

Figure 2 – Results from the study of latent learning in rats [Tolman and Honzik, 1930].

Tolman believed every time the rats ran through the maze they were learning about the correct path, even if they weren't given a reward upon finishing the maze. According to Tolman, the rats weren't finishing the maze quickly and with few mistakes because they had no good reason to do so, but as soon as they began receiving a reward they used the knowledge acquired in the ten previous days to complete the maze faster and get to the cheese as soon as possible. Tolman stated the rats had developed cognitive maps, spatial representations of the maze [Tolman, 1948].

Tolman's work, as well as the work of others, challenged behaviorism by showing learning could take place without reinforcement. It also demonstrated that thinking (cognitive maps in Tolman's experiments), unlike what Watson and Skinner believed, has a key role in at least some forms of learning [Tolman, 1948].

Observational learning is a form of latent learning whereby one learns by watching others. Behaviorists couldn't explain why children sometimes wouldn't repeat behavior that had been reinforced, nor could they explain why they would imitate behavior weeks after the initial observation without being subjected to any kind of reinforcement. It so happens that we often learn by observing models, such as parents, teachers or someone we admire and look up to. It's a kind of learning that doesn't require reinforcement and is therefore considered a form of latent learning. With observational learning we learn by observing others and the consequences (reinforcement) of their behaviors, it spares us the need to experience the consequences ourselves [Bandura, 1977].

Albert Bandura, together with Dorothea Ross and Sheila A. Ross showed how children can imitate aggressive behavior [Bandura et al, 1961]. They had two groups of preschool boys and girls watch an adult interact with a large doll. Children from the first group watched the adult ignore the doll and play quietly with other toys, whereas children from the second group observed the adult be physically and verbally aggressive with the doll. The children were then introduced to a room with several toys, but were moved to a different room as soon as they began playing. This was done on purpose in order to cause them frustration. The next room contained a doll just like the one the children had watched the adult playing with. Bandura and his colleagues observed that the children who had seen the adult play quietly, played quietly as well. The other children however, imitated the adult's aggressive behavior and punched, kicked and insulted the doll just as he had. This behavior was fueled by the frustration of having been removed from the room with toys [Bandura et al, 1961].

2.1.2 Cognitivism

Cognitivism doesn't contradict behaviorism, it extends it. Behaviorists, such as Watson, believed thinking took no part in the learning process. Thinking couldn't be observed and therefore shouldn't even be considered [Watson, 1913]. Skinner on the other hand didn't deny thinking, but viewed it merely as another behavior [Skinner, 1953]. Cognitivists accept conditioning and response to stimulus, but believe an individual's response to it depends on what the stimulus means to the individual. The different interpretations of the same stimulus made by two individuals explain why they may react differently to it [Good and Brophy, 1990].

For cognitivists, learning involves the reorganization of the brain's cognitive structures. These structures are internal knowledge representations which can be augmented, combined or altered to create new knowledge representations. With cognitivism, both knowledge and memory have key roles in the learning process [Good and Brophy, 1990]. There are several theories within Cognitive Psychology that explain how learning takes place. One of the most popular ones is the Cognitive Information Processing Theory, which originally used the way computers work as a metaphor to explain how the human mind processes information. The process by which learning occurs involves the passage of newly acquired information from input sensors to long-term storage to be retrieve when necessary. In 1968, Atkinson and Shiffrin devised a model that describes the human memory as being composed of different stages of storage [Atkinson and Shiffrin, 1968]. In the human mind, memory is the system that processes new information in a three step operation. First, a sensory register receives information from the senses; this process never lasts longer than four seconds. All received information is analyzed and pertinent information is transferred to short-term memory. The majority of information that reaches the sensory register never makes it to short-term memory. Information that does reach short-term memory can stay there for twenty seconds or more if trained repeatedly. Short-term memory can keep between five and nine items at a time, which is why phone numbers usually have between seven and nine digits. One way of increasing the capacity of short-term memory is dividing information into smaller, meaningful parts. Long-term memory is where short-term memory information is stored for long term use. There’s no limit to the amount of information it can store. Information can reach long-term memory by establishing meaningful relations with previously existing information or by repeated memorization [Atkinson and Shiffrin, 1968].

2.1.3 Constructivism

Constructivism as a learning theory states that learners construct new knowledge based on past experiences, mental structures and beliefs. Each individual has a unique view of the world, which influences the construction of new knowledge and new realities. According to constructivists, this singular perspective of reality means each individual must be actively involved in constructing knowledge for himself instead of simply being a passive learner. By being actively involved in his own learning process the individual can learn more deeply than if he was reading from a book, for example. Learning, according to this theory, must take place in a real-world scenario and the learner must be given the freedom to explore and construct knowledge for himself, while the teacher must function as a guide or mentor and not as someone who transmits knowledge [von Glaserfeld, 1989][Bauersfeld, 1995][Gamoran et al, 2000].

Jonassen [Jonassen, 1994] tackles an interesting problem: if each individual constructs its own reality, having a unique perspective of the world, then how can we as a society coexist and communicate? Considering that, apart from the reality each of us constructs in our mind, there is a physical world subject to universal physical laws that we all perceive in a similar manner, then our individual realities all have a common foundation. In addition, constructivists argue that we share our realities through a process of social negotiation [Jonassen, 1994].

In constructivism, the individual's culture and prior experiences are of great importance to the learning process. Since individuals are encouraged to discover knowledge on their own, using a hands-on, realistic approach to construct their own knowledge, a process which constructivists believe is heavily influenced by the individual's belief system and prior experiences, those experiences and the individual's cultural background play a very important role. Language, history and mathematical systems are very much culture specific and influence the way the learner views the world. In this way, social interaction too is relevant, since the learner depends on social exchanges with more knowledgeable individuals to acquire those culturally-specific symbols and knowledge [Wertsch, 1985].

Unlike what is seen in traditional practices of education, constructivism stresses the learner should be more actively involved in the learning process and be more responsible for his own knowledge acquisition. The teacher should play a secondary role in the learning process, acting as a counselor that steers the learning process in the right direction, but lets the student be free to discover and create his own knowledge. In this sense, the teacher must adapt to a scenario in which he should no longer teach in the traditional manner involving the transmission of knowledge to students, but should instead assist students in their own understanding of the content. This change in roles has important implications because, as a mentor, an instructor must possess different skills than as a teacher. The mentor must create an environment where students can learn for themselves, he must dialogue with and support his students, and instill in them curiosity, creativity and a critical mind. In contrast, the traditional teacher usually transmits knowledge and gives answers while the student passively listens [Bauersfeld, 1995][Gamoran et al, 2000][Brownstein, 2001].

Cooperation is also important in the learning process as students exchange ideas and perspectives, and learn from each other. The aim is to work as a team, sharing goals, responsibility and honing important social skills, and not work against each other [Duffy and Jonassen, 1992]. According to constructivism, the learner's motivation is vital. Motivation is correlated to the student's confidence in his ability to learn and solve new problems [von Glaserfeld, 1989]. If the student manages to successfully solve challenging problems, he'll be far more confident and motivated than if he were to receive (possibly unfounded) praise. In this way, ensuring the student is properly and continuously challenged is one of the mentor's functions. The mentor should guide the learning process so that it's always one step ahead of the student's current level [Prawat and Floden, 1994][Brownstein, 2001].

For constructivists, the knowledge one attains must give one the ability to solve real-world problems. In order for this to happen, said knowledge must be placed into context. By learning in the appropriate context, the learner will be faced with the complexity of the environment and will learn about which concepts to apply and when to solve context-related tasks. The idea is to accustom students to authentic settings and problems by having them explore, learn and solve problems in authentic environments, or environments that mimic real ones [Duffy and Jonassen, 1992][Brown et al, 1989].

Assessment in Constructivism isn't a separate procedure, but is instead a part of the learning process that must be performed continuously. When assessing a student's knowledge, the mentor must engage him in a conversation in order to determine his current level of knowledge. This in turn will allow him to suggest ways of improving and steer the learning process in the right direction. Self-assessment and peer assessment are also important, and it's suggested that student's exchange evaluations of each other's work [Holt and Willard-Holt, 2000].

2.2 Learning Styles

Learning styles represent models of how a person learns best. Students learn in different ways and depend upon many different and personal factors [Ritu and Sugata, 1999]. Initially, the idea was that each individual had a single learning style. However, more recent studies have shown the majority of people are actually multimodal, meaning they have more than one learning style [Fleming, 2007][Miller, 2001]. Learning styles have many proponents and several studies support their use [Kolb and Kolb, 2005][Montgomery and Groat, 1998][Richmond and Cummings, 2005], but they have also been criticized by many [Brown et al, 2007][Stahl, 2002][Hargreaves et al, 2005]. There doesn't seem to be, however, evidence suggesting the use of learning styles is detrimental.

Several models of learning styles exist, but two of the most well-known are David Kolb's experiential learning model [Kolb, 1984] and Neil Fleming's VAK model [Fleming and Mills, 1992].

2.2.1 David Kolb's Experiential Learning Model

David Kolb, with the help of his associate Roger Fry, developed a theoretical model of the process of experiential learning. As the name indicates, this model focuses on learning through experience, but it also hints at its origins in the works of Jean Piaget, Kurt Lewin and John Dewey [Kolb, 1984][Smith, 2001]. According to Kolb, the model reflects the transformation and understanding of experience into knowledge. The model is comprised of four abilities, concrete experience (feeling) and abstract conceptualization (thinking), which represent two forms of understanding the experience, and reflective observation (watching) and active experimentation (doing), which are ways of transforming experience into knowledge. Learning may begin at any stage, although it's recommended that it begins with an experience, but once it does it should follow the cycle sequentially. Figure 3 shows how concrete experience is followed by observation and reflection, which leads to the formation of abstract concepts about what has been observed. Finally, those abstract concepts can be tested in new situations and serve to create new experiences [Kolb, 1984][Smith, 2001][Chapman, 2009].

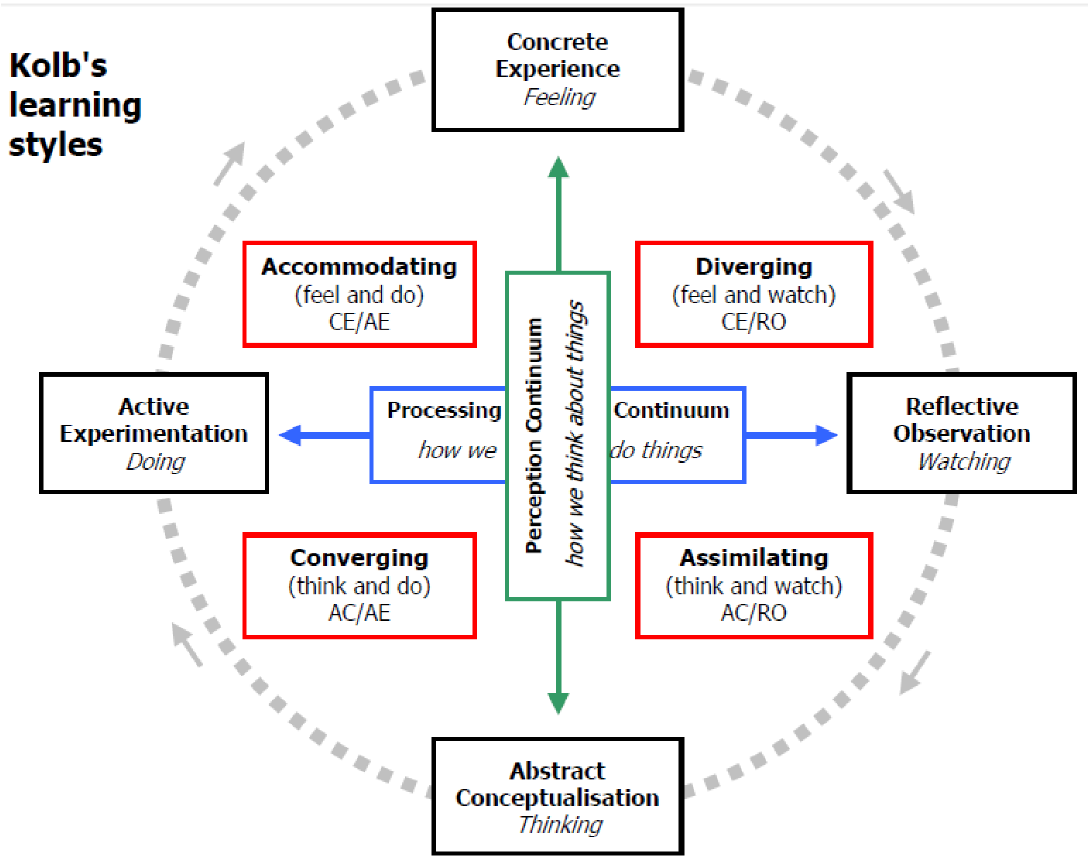

Figure 3 – David Kolb's Experiential Learning Model [Chapman, 2009].

Ideally, in order to learn one should be in possession of all the abilities described in the model. However, in reality we tend to choose one form or another of comprehending an experience and turning it into knowledge. Some of us gain understanding by experiencing concrete situations and trusting our senses, whereas others do so by thinking about and analyzing the situation. When turning experience into knowledge, some prefer to observe the circumstances and reflect upon them, while others choose to put their ideas into practice and use the results to gain knowledge. Taking this into consideration, Kolb created the Learning Styles Inventory to determine an individual's learning style, and with basis on the results and his observations, identified four different learning styles: converger, diverger, assimilator and accommodator [Kolb, 1984][Smith, 2001][Chapman, 2009]. Table 1 illustrates how each learning style is a product of the different ways of 'grasping an experience' and 'transforming that experience' into knowledge.

Table 1 – Kolb's Learning Styles Inventory.

	Doing (Active Experimentation - AE)	**Watching (Reflective Observation - RO)**
Feeling (Concrete Experience - CE)	accommodating (CE/AE)	diverging (CE/RO)
Thinking (Abstract Conceptualization - AC)	converging (AC/AE)	assimilating (AC/RO)

The converger's primary learning abilities are abstract conceptualization and active experimentation. Convergers are good at putting ideas into practice and tend to solve problems by reasoning about them and then testing their theories. They tend to be unemotional and aren't particularly interested in social issues. People with this learning style have a greater vocation to technology careers [Kolb, 1984][Smith, 2001][Chapman, 2009].

The diverger possesses the abilities concrete experience and reflective observation. These people are good at coming up with ideas and thinking "outside the box". They have a well-developed imagination, are sociable, artistic and are interested in a wide range of cultural issues [Kolb, 1984][Smith, 2001][Chapman, 2009].

The assimilator tends towards abstract conceptualization and reflective observation. When learning, people with this learning style like to read, analyze theoretical models and reflect on the subject. They're good at understanding ideas and situations in a logical manner, and are thus apt at creating theoretical models. Assimilators are more interested in knowledge than in other people and usually do well in scientific careers [Kolb, 1984][Smith, 2001][Chapman, 2009].

The learning abilities of accommodators are concrete experience and active experimentation. Accommodators learn well from experience and are very practical people. They tend to rely more on their emotions than on logical reasoning and solve problems intuitively. They're best suited for careers which require quick reactions and prefer working with others [Kolb, 1984][Smith, 2001][Chapman, 2009].

Kolb's model works as a set of guidelines and learners aren't and shouldn't necessarily be restricted to only one learning style. People do tend to show a strong preference for one style over the others, and will learn more easily if the learning process suits that particular style. For example, an assimilator might feel lost if given a task without any accompanying information, whereas an accommodator might feel frustrated if he has to spend a lot of time reading instructions before being able to engage in the task at hand [Smith, 2001][Chapman, 2009].

2.2.2 VAK Model

VAK stands for Visual, Auditory and Kinesthetic. The VAK model focuses on sensory modality as a learning style dimension, a concept substantiated by the literature on neuro-linguistic programming that concerned itself with the different forms by which information is perceived (visual, aural and kinesthetic) [Fleming and Mills, 1992]. It describes learning preferences in a simple manner, and provides tools and methods to assess and understand a person's dominant learning style that are easy to use. This inherent simplicity is one of the key reasons behind the model's popularity [Chapman, 2003][Chapman, 2005]. According to the VAK model, people tend to have a dominant learning preference, but they might also be multimodal. The result of the VAK questionnaire assigns a score to each modality. A person is considered multimodal if no single modality scores markedly above the others. Multimodal people fall into two categories: type one and type two. Type one people are flexible and context specific, they choose the modality that better fits the learning situation. For example, when taking part in a workshop a (type one) multimodal person will express her kinesthetic preference, but when attending a lecture the preference expressed will be auditory. Type two learners prefer to acquire knowledge using all of their preferred sensory modes. Obtaining information this way takes longer, but the learner frequently exhibits a deeper and broader understanding of the subject, which in turn might result in better decision making [VARK, 2015].

A person's dominant learning style within the VAK model can be assessed by having that person fill in a specific questionnaire. The questionnaire is then used to determine the individual scores for each of the possible learning styles. The model also provides guidelines that can be used to design learning methods and/or experiences that best suit each preference [Chapman, 2003][Chapman, 2005][VARK, 2015].

Learners that display a strong preference towards visual learning prefer information that is visually represented (can be seen), such as pictures, diagrams and videos. They tend to create mental images when thinking, and retain information better when they are presented with visual representations of it. Visual learners usually possess strong visual skills and are therefore apt at building puzzles, reading maps and charts, etc [Chapman, 2003][Chapman, 2005][VARK, 2015].

The auditory learning style consists in a preference for information that can be heard. People with this preference learn better when they attend lectures where they can hear others speak, but also by having group discussions, listening to the radio and audio files, reading out loud, as well as talking things through (including talking to themselves). Auditory learners are frequently good at oral communication [Chapman, 2003][Chapman, 2005][VARK, 2015].

The word "kinesthetic" comes from the Greek words "kīneîn", which means "to move", and "aisthesis", which means "perception". As the origin of the word implies, it describes the perception of information through movement, specifically movement of the muscles and tendons. The kinesthetic learning preference is therefore related to practical, physical experience, and kinesthetic learners (supposedly) learn better by interacting with the world around them. Activities suited to kinesthetic learners include engaging in case studies, practical activities (experiments, exercises) or working examples, taking notes or annotating text while reading, etc [Chapman, 2003][Chapman, 2005][VARK, 2015].

2.3 Adaptive Hypermedia Systems

Hypermedia is a term derived from hypertext and refers to hyperlinks, text elements and other multimedia elements, mingled together to form a complex, non-linear and interactive network of information. Both terms were first introduced by Ted Nelson in 1965 [Nelson, 1965].

Adaptive hypermedia first appeared in the beginning of the 1990s, as a result of the interaction of two well-developed fields of research, Hypertext and User Modeling. By then, researchers were aware of the limitations of static hypertext and were trying to find ways of adapting their systems to the characteristics and needs of different users. Achieving this required creating a model of the user and his goals. The user modeling community not only provided the necessary assistance to accomplish this task, but also helped separate research teams find each other and was crucial in fostering adaptive hypermedia as a new branch of user modeling research [Brusilovsky, 2001].

Unlike conventional hypermedia systems, which use a one-size-fits-all approach, the main purpose of adaptive hypermedia systems (AHS) is adapting interface, content presentation, link navigation and so on, to the specific characteristics, needs and interests of different users. As these goals and characteristics change, so does the content presented by the system [Brusilovsky et al, 2007] [De Bra, 2006].

After 1996, research into adaptive hypermedia grew very rapidly. The field had reached a state of maturity that allowed new research to be based on and expand previous research. Many of the systems developed after 1996 were real world systems, whereas the first systems developed were laboratory ones. Another reason for the progress of adaptive hypermedia research was the widespread use of the World Wide Web. The diversity of users on the Web emphasized the need for adaptability and opened up new venues of research [Brusilovsky, 2001]. Nowadays, in particular, adaptation is proving helpful in dealing with the excess of information available on the Web [Knutov et al, 2009].

There are several types of AHS, namely adaptive educational hypermedia systems, online information systems, information retrieval hypermedia, online help systems, institutional hypermedia and systems for managing personalized views. Most systems developed after 1996 were AEHS and online information systems. There were also a fair number of information retrieval hypermedia systems, whose scope was extended to include systems for managing personalized views. The remaining types haven't been given much attention ever since. The majority of systems developed in the past few years have been adaptive educational hypermedia systems [Brusilovsky, 2001].

Online information systems encompass such systems as handheld guides, adaptive decision support systems and virtual museums, as well as more generic online information systems. The greater a system's level of specialization, the easier it is to adapt to the user's interaction with the system.

The adaptation provided by a handheld museum guide benefits not only from a restricted and structured hyperspace of objects, but can also be enhanced if the user's position and behavior is ascertained by the system. Such a system is capable of adapting not only to the user and his interaction with the system, but also to his interaction with the physical world.

Adaptive decision support systems use expert knowledge and artificial intelligence to assist in the solution of particular problems. By combining information about context and goals with specialized information about the domain, these systems are capable of creating detailed user models and providing adaptation of higher quality [Brusilovsky, 2001].

Adaptive educational hypermedia systems place the focus on helping users achieve their learning goals. For this purpose, characteristics such as the user's knowledge and learning style are particularly important [Chepegin et al, 2004][De Bra, 2006]. The increase in AEHS being developed after 1996 was related to the rapid spread of the Web. Offering education over the Web was an interesting option that led to the creation of several systems. These systems vary greatly, and many have been developed to be full-fledged frameworks and provide different authoring tools that facilitate the creation of content.

The interest in e-learning has recently been fueled anew by the emergence of massive online open courses. AHS have been the subject of much research but more development, experimentation and implementation are necessary to conclude about the adequate features and effectiveness of these systems [Knutov et al, 2009][Fernandes et al, 2012].

A common architecture for AHS is the one proposed by Benyon [Benyon, 1993] and De Bra [De Bra et al, 2004] which consists of three interdependent modules:

- The User Model (UM) may contain personal information about the user, as well as data concerning his interests, goals, et cetera. In the case of AEHSs the Student Model must consider domain dependent data and domain independent data.
- The Domain Model (DM) consists of a semantic network of domain concepts. Its main purpose is providing a representation structure for the user's domain knowledge.
- The Adaptation Model (AM) represents and defines the interaction between the user and the application. It displays information to the user based on the user model and updates its information according to the user's actions.

By relating the user model to the domain model the system can, through the adaptation model, adapt its content, navigation and interface to each user's specific needs. Triantafillou

[Triantafillou et al, 2003] underlined the importance of adaptability in educational hypermedia, considering these systems are meant to be used by several learners.

2.3.1 The User/Student Model

AHS change several aspects of the system based on the user's characteristics, such as goals and preferences, as well as usage data. These characteristics, which can be provided by the user or inferred by the system, are stored in the user model [Brusilovsky, 2001]. Some of the properties kept in the user model are static, while others are dynamic. An AHS must be capable of using these properties and keeping them up to date, by observing and analyzing the user's actions [Knutov et al, 2009]. In the case of AEHSs, the user model, or Student Model, also stores the user's knowledge. The purpose of AEHSs is helping users achieve their learning goals. When one goal is reached, the system re-adapts to the newly acquired knowledge [Brusilovsky, 2001][Martins et al, 2008b]. This means that the student model is of particular importance for AEHS because the information it contains about the user's knowledge is crucial for a properly adapted learning experience.

The student model (Figure 4) includes domain dependent data and domain independent data. The first consists of the user's domain knowledge, learning goals and data related to the user's interaction with the system, such as a complete description of his navigation through the course. Most commonly, the domain dependent data is related to the domain model through an overlay structure [Knutov et al, 2009].

Domain independent data consists of personal information, demographic data, academic background, qualifications, cognitive capacities, etc. The domain independent data has a two element structure, the psychological model and the generic model. Psychological data is related to the student's cognitive and affective aspects. Some studies have shown the quality of interaction is affected by the differences between cognitive abilities and personality traits. Since these features are usually permanent, the system is capable of knowing in advance to which characteristics it must adapt. The generic model stores information about the user's interests, general knowledge, background, etc [Kobsa, 1993][Martins et al, 2013].

Depending on the system being developed, some of these features are relevant for the user model and some are not [Brusilovsky, 2001][Martins et al, 2008a][Brusilovsky, 1996a]. Determining which of the user's characteristics should be used is an important step in the creation of an AHS [Martins et al, 2008b].

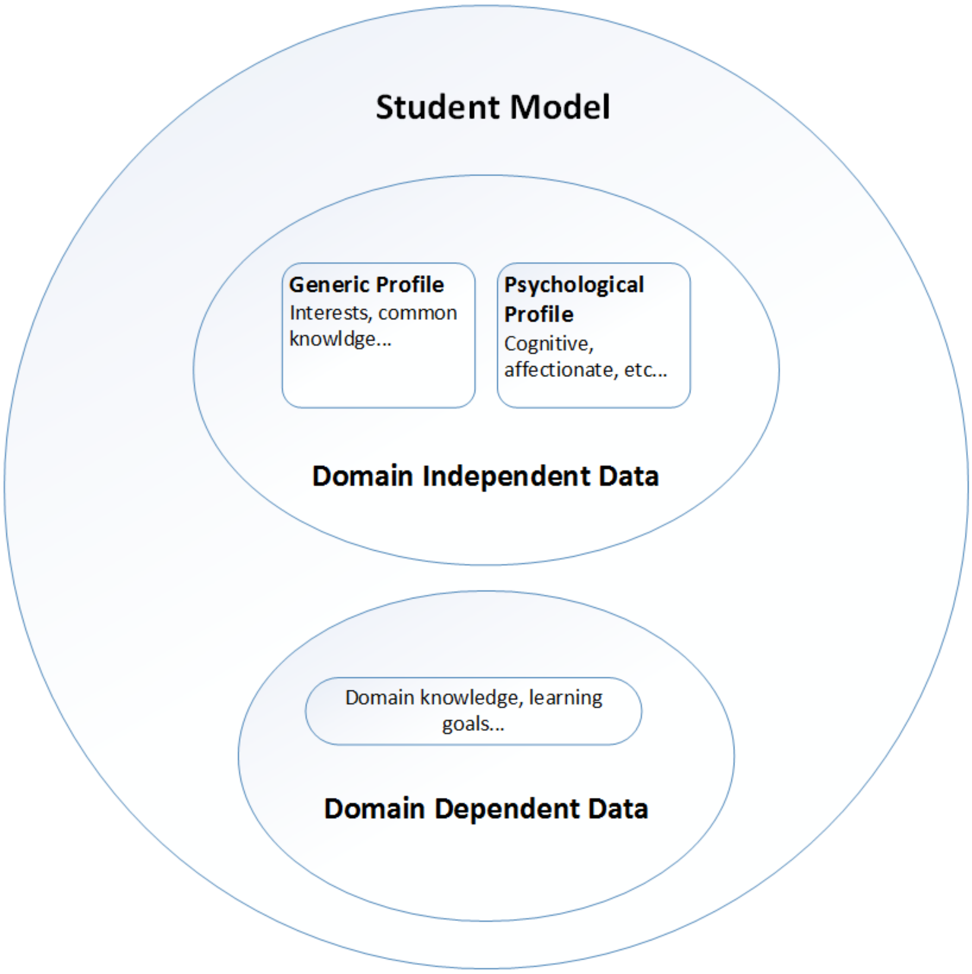

Figure 4 – The student model.

The student model is implemented by means of two different techniques: knowledge-based adaptation and behavioral-based adaptation [Martins et al, 2008a]. For knowledge-based adaptation, a set of initial heuristics are produced with information about the user obtained by means of questionnaires and/or observation. Behavioral-based adaptation comes from monitoring the user during his activity.

Behavioral-based adaptation is implemented using the overlay and perturbation methods. With the overlay method the user's knowledge is directly mapped over the domain model's knowledge space [Knutov et al, 2009][Martins et al, 2008a]. In this way, the user's knowledge is a subset of the domain knowledge. For each concept in the domain model, a set of attributes, which may be binary, qualitative or quantitative, is associated with the user's knowledge [Knutov et al, 2009][Martins et al, 2008a]. The overlay method requires the domain model to represent individual topics and concepts. The complexity of this method depends on the granularity of the domain model and on the system's assessment of the user's knowledge. The overlay method results in a very flexible user model that can be used in several different domains [Martins et al, 2008a].

Stereotypes may be combined with the overlay method. As the name implies, stereotypes are groups represented by a set of pre-defined characteristics. Users classified into a stereotype inherit that group's characteristics. The choice of features for this classification depends on the desired degree of granularity [Martins et al, 2008a]. When combining stereotypes with the overlay method, the system begins by classifying the user into one of the defined stereotypes and gradually changes it as the user interacts with the system [Brusilovsky, 2001][Martins et al, 2008a].

The perturbation method consists in keeping a model of possible errors for each knowledge concept. With this method, the user model stores the errors committed by the user. Unlike with the overlay method, with the perturbation method the user's knowledge represents a perturbation of the domain knowledge. The system applies the perturbation model to the domain knowledge space and uses induction to infer the student's knowledge level [Martins et al, 2008a].

2.3.2 The Domain Model

The domain model is a conceptual definition of how the domain is structured. It frequently consists of concepts and the relationships between them. Concepts are abstract information items that represent the expert's knowledge in the domain application [Wu et al, 2000].

The relationships between concepts and the way in which these are presented are kept in this model. The DM doesn't contain everything about the application, but contains the information needed by the system in order to know what needs to be adapted and how concepts are represented in terms of content. The domain model is an integral part of an AHS.

The DM has several functions. All the conclusions taken from the interaction between the user and the system are based on the domain model. Consequently, the DM must be abstract enough to allow the necessary inferences to be made. It's also in the DM that the adaptive features of the system are defined by describing alternative representations of domain characteristics. Another purpose of the DM is storing the system's measurable features so their effectiveness may be verified against the necessary rules. Finally, the DM is also the base for the user model. The domain model contains a description of the system, which is used by the system to register in the user model the data obtained from the user's interaction with the application [Benyon and Murray, 1993].

A cognitively valid domain model should contain descriptions with three different levels of abstraction, namely the task level, the logical level and the physical level. The structure of the domain must be described in the DM for each of these levels. The task level is intended to let the user know about the system's purpose. A task level description should tell the user what the system's supposed to do but not actually do it. A logical level description, on the other hand, explains how something works. It does so by describing what actions are undertaken and which objects manipulated. The physical level is concerned with what must be done to perform a certain job. It presents the succession of tasks needed to achieve that and is related to the presentation of the system, dialogue control and physical actions. These three levels of abstraction are present in the DM together with descriptions of the mappings between them. A task corresponds to one or more logical procedures, which in turn correspond to several physical actions [Benyon and Murray, 1993].

The concepts of the DM can have distinct functions, weights and meanings in different AHS. However, for most AHS each concept is connected with/related to other concepts, representing a semantic net [Martins et al, 2013]. Concept relationships represent sequences of two or more concepts. Concepts have types, the most common one being the hypertext link. There may also be other types of relationships, related to the system's adaptation. One such type is the prerequisite. This type doesn't imply the need for a link from one concept to the other. It means the user should first view the content related to concept A1 before moving on to concept A2. The system is made aware of this requirement through the existence of the prerequisite type. Link adaptation is the technique used to inform the user whether it's appropriate or not to click on a particular link [Wu et al, 2000].

2.3.3 The Adaptation Model

The adaptation model represents and defines the interaction between the user and the application. It displays information to the user based on the user model and updates its information according to the user's actions. By relating the user model to the domain model the system can, through the adaptation model, adapt its content, navigation and interface to each user's specific needs.

An adaptive hypermedia system can adapt to several characteristics of the user. These features may change from user to user and may even be different for the same user in different times and situations. The user features usually taken into account by AHS are goals, knowledge, background, hyperspace experience and preferences [Brusilovsky, 1996b].

The user's knowledge of the subject is generally the most important characteristic for an AHS and its value varies as the user interacts with the system. For this reason, the system must monitor the knowledge variable and update its value in the user model. This user feature is used by the majority of presentation techniques [Brusilovsky, 1996b].

A user's goals are related to what the user intends to do with the system, namely what the user is using it for and why. These goals are context-dependent and may change as the user uses the system. The type of hypermedia system itself influences the user's goals. Information retrieval systems are used to perform searches, application systems may be used for work, and educational hypermedia systems are used with learning as a goal. Goals can be high-level or low-level with certain systems allowing for both. Low-level goals change more frequently than high-level goals. For example, the user of an AEHS has the high-level goal of learning, but as his interaction with the system progresses his low-level goals will change from learning one specific concept (or set of concepts) to learning another, and from performing a given activity to performing a different one [Brusilovsky, 1996b].

The user's background refers to knowledge and experience the user possesses, that was acquired outside the context of the hypermedia system but is pertinent to its usage. It might include professional experience, educational background and the user's perspective [Brusilovsky, 1996b].

The experience the user has with the system's hyperspace structure is a feature that must be taken into account. The user might be acquainted with the hyperspace structure while having little knowledge of the subject, or he might be familiar with the subject but have no experience with the hyperspace. This feature is commonly used to tailor each user's navigation through the hypermedia system, but can also be used to adapt the interface and related assistance [Brusilovsky, 1996b].

User's preferences are related to the fact that a user might prefer a certain link, page, or page section over another. These preferences can't be deduced by the system; therefore the user has to let the system know about them. This can be done directly, using a questionnaire for example, or indirectly by receiving feedback from the user. User's preferences are used mostly by information retrieval systems [Brusilovsky, 1996b].

Adaptive hypermedia normally involves two types of adaptation: content-level adaptation and link-level adaptation. The first one is called adaptive presentation and is achieved by adapting the content of a system's hypertext documents. Adaptation of the links connecting hypertext documents to one another, as well as adaptation of index pages and maps consists in link-level adaptation and is referred to as adaptive navigation support (Figure 5) [Brusilovsky, 1996b].

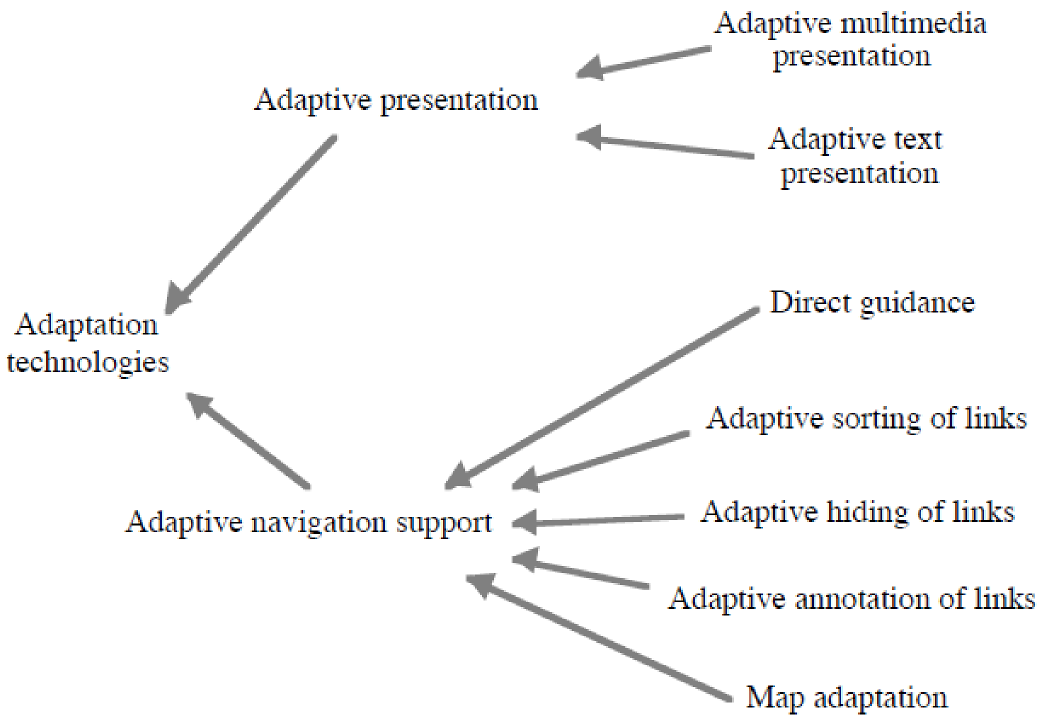

Figure 5 – Adaptation techniques in adaptive hypermedia [Brusilovsky, 1996b].

2.3.3.1 Adaptive Presentation

The purpose of adaptive presentation techniques is adapting a web page's content to each user's particular characteristics, goals and knowledge. In this sense, a user with more advanced qualifications might be shown more complex and complete information; whereas an inexperienced user might be presented with information structured in a simpler manner and be given additional explanations. This content commonly takes the form of text, but it can also be represented by multimedia objects.

There are several techniques used to adapt content in hypermedia: conditional text, stretchtext, page and fragment variants, and frame-based.

With the conditional text technique all text content related to a concept is broken into several pieces. A condition related to the user's knowledge is associated with each piece of text. When showing content related to a concept, only the pieces of text whose condition is true will be shown. This is a low-level technique which entails some work on the part of the author but is also quite flexible [Brusilovsky, 1996b][Hill and Carver Jr., 2000].

A similar but higher level adaptation technique is stretchtext. This technique also works by showing or hiding different portions of content depending on the user's knowledge, but uses a special kind of hypertext called stretchtext to achieve it. Stretchtext resembles a hyperlink in the sense that it provides a more descriptive explanation of something, but unlike a hyperlink it does so by expanding or contracting that explanation without redirecting the user to a new webpage. With this technique the system collapses content which is unsuitable to a specific user given his current level of knowledge, and uncollapses content which is pertinent. This technique also allows content to be customized by the user, who can collapse and uncollapse stretchtext as he pleases. This gives the system the ability of adapting to both the user's knowledge and the user's preferences, because it can monitor the user's behavior as he collapses and uncollapses text and update the user model accordingly. An example of adaptive presentation using stretchtext would be showing a user with little knowledge about a concept a page with additional explanations uncollapsed, and sections containing more detailed and complex information collapsed. Should the user choose to expand the hidden sections, the system would take that choice into consideration when next showing the user information about that concept [Brusilovsky et al, 2007].

The page variants technique is the simplest solution available for adaptive presentation. Adaptive Hypermedia systems that use this technique employ user stereotypes and keep several variants of the same page, each adapted to a different stereotype [Brusilovsky et al, 2007].

With the fragment variants technique each page is composed of multiple fragments and each of these fragments can contain different presentations of the same content. This technique is similar to the page variants technique, but instead of keeping different variants of one page, it keeps variants of concept explanations.

The frame-based technique associates each concept with a frame. A frame has several slots that contain links to other frames, different content presentation, activities, and so on. Which of these slots are shown to a user and in which order depends on the chosen presentation scheme. This choice is determined by the activation of presentation rules, which can be defined by any characteristic expressed in the user model.

There are also different methods of adapting content in hypermedia. These methods are implemented using the techniques described above.

Additional explanations refers to a method that consists in hiding from the user information about a concept which he isn't yet prepared to see, and showing him content that is relevant given the feature values in his user model. For example, additional content explanations can be presented to a user whose knowledge level is low, but might be kept hidden from a user with greater knowledge. On the other hand, the more advanced user might be interested in low level details that simply aren't appropriate for the beginner user. In the same manner, the system might show a user content that fits his goals, but hide it from a user with different goals. This method can be implemented using the conditional text, stretchtext and frame-based techniques [Brusilovsky, 1996b].

The prerequisite explanations method takes into account the user's knowledge of related concepts. When presenting the user with information about a concept, if his knowledge of a prerequisite concept (or concepts) is below a certain level the system will first show information about that concept. The conditional text and stretchtext techniques can be used to implement this method. The frame-based technique can also be used if the appropriate conditions are set for the knowledge level of related concepts [Hill and Carver Jr., 2000].

Similarly, the comparative explanations method also considers how much a user knows about related concepts. In this case, if the user has already acquired knowledge about a concept similar to the one the system is about to present, it shows a comparison between the two concepts, highlighting the differences and similarities between them and making it clearer why those concepts are similar but not the same. This method can be implemented using the same techniques as the prerequisite explanations method [Hill and Carver Jr., 2000].

Explanation variants is a method that was created under the assumption that hiding and showing information about a concept doesn't provide enough adaptation to each user's characteristics, because different users may learn in different ways and require different content. With this method the system keeps several variants of the same content, and presents the user with the one that is best adapted to his user model. This method may be implemented using the page variants, fragment variants, conditional text and frame-based techniques [Brusilovsky et al, 2007].

The sorting method consists in organizing the different fragments of information displayed on a page by placing those that are more relevant to the user, given his knowledge and background, at the top of the page. It can be implemented by the frame-based technique [Brusilovsky et al, 2007][Hill and Carver Jr., 2000].

2.3.3.2 Adaptive Navigation Support

The main purpose of adaptive navigation support is assisting and guiding users as they navigate the hyperspace. This can be achieved by adapting hyperlinks to the user's knowledge, goals, as well as other features present in the user model. Adaptive navigation support can be accomplished by using one or more of the following five techniques: direct guidance, sorting, hiding, annotation and map adaptation. These techniques may be used alone or combined [Brusilovsky et al, 2007].

Direct guidance, as the name suggests, guides the user through the hyperspace in a straightforward manner. It does so by presenting the user with the link best adapted to his individual characteristics. Such a link is presented in a very direct manner and the user isn't given an alternative. He must either follow the link or not. This "all-or-nothing" approach has the advantage of reducing the cognitive load placed on the user, but it's also the reason why this technique provides limited support. If the user chooses not to follow the link, no further assistance will be given. Moreover, since the system will receive no feedback from the user it won't be able to adapt to his progress. For such reasons this technique shouldn't be used alone, but should instead be combined with other techniques [Hill and Carver Jr., 2000].

Sorting consists in ordering all links in a page according to their importance to a given user. This technique considers the user's characteristics to determine which links are more relevant and emphasizes those links by presenting them at the top of the page, while placing the less relevant links closer to the bottom of the page. The disadvantage of this technique lies in the fact that, as the user model changes, the relevance of a page's links might change as well. This means the same page could have a different link order every time the user visits that page. This technique shouldn't be used with links whose order can't be changed, such as indexes and tables of content [Brusilovsky, 1996b][Hill and Carver Jr., 2000].

Hiding is the technique most widely used in adaptive hypermedia. With this technique the system guides the user in the appropriate direction by hiding the links that aren't suitable for his knowledge level, goals or other characteristics featured in the user model. Hiding is a technique that is easy to implement and offers a simple and effective way of helping the user navigate the hyperspace. Because the hyperspace is simplified it's easier for the user to progress without being overwhelmed by the excess of information and variety of paths. When compared to sorting, this technique also has the advantage of showing the user pages whose structure is more permanent. However, as is the case with direct guidance, hiding is a technique that employs an "all-or-nothing" approach. This technique classifies links as either being suitable or not suitable, and shows only those that are suitable. As with direct guidance, the user can only follow or not follow a link. As such, there's no way for the system to receive feedback from the user about its effectiveness and update the user model according to that information [Wu et al, 1998][Brusilovsky et al, 2007].

The idea behind adaptive annotation is that links can be enhanced by text or visual hints meant to provide the user with more information about a link. For instance, if a link's font is red that might signify the content the link points to isn't suitable to the user. Other forms of annotation may be used, such as different font sizes and types, and different icons. The annotation used changes according to each user model. With this technique users are free to choose which links they want to follow and in what order. The annotations will indicate whether a link is relevant or not, but the user isn't forced to follow a strict path and can progress as he sees fit. This has the advantage of letting the system adapt to the user's behavior. This technique has wide applicability and allows the structure of pages to remain stable. It's not as simple as hiding *or* direct guidance, but it's intuitive and prevents the formation of wrong mental maps. Hiding may be simulated by dimming links which would otherwise be hidden. Users can still follow these links, but the cognitive overload is somewhat diminished [Wu et al, 1998][Brusilovsky et al, 2007].

Map adaptation is a simple technique that consists in adapting the links in an image to different users. Users see the same image, but the links associated with the image components will be adapted to the features in each user model, and as such each user will be directed to a different page after clicking in the same component. This technique is a more visual version of direct guidance; therefore it presents the same disadvantages [Brusilovsky, 1996b].

2.4 Adaptive Hypermedia Architecture

AHA! is an open source adaptive hypermedia system developed with the intention of being a general purpose adaptive hypermedia architecture. It is considered an example of simplicity when compared to other AHS. The system was written in Java and uses Java Servlets. The first version of AHA! was released in the year 2000. Its third and current version was released in 2007 [De Bra et al, 2003].

AHA! comprises the following adaptive hypermedia methods and techniques:

- The system's user model is made up of concepts. Each concept can have many attributes, which may be Boolean values, integer values or strings. Whenever a user views a page that information is passed on to the adaptation engine and the user model is adapted accordingly. For example, a page view might increase the value of the knowledge attribute of one or several concepts. This change will be reflected in the user model [De Bra et al, 2003].
- AHA! uses link hiding and link annotation as a way of providing adaptive navigation support. The author defines a priori whether a page is suitable or not for a given user, by constructing a Boolean expression with values from the user model. When a user visits a page, if the suitability expression of a link on that page evaluates to *true*, the link is shown in blue if it's never been visited or in purple if it has already been visited. If the expression evaluates to *false*, the link is shown in black and isn't underlined. In this way, links not suitable to a certain user will be concealed. However, it's possible to change the color scheme to make all links visible, albeit with different colors [De Bra et al, 2003].
- Content adaptation is achieved through the use of the conditional inclusion of fragments technique (corresponding to the conditional text technique). However, in AHA! this technique can be used not only with fragments of text, but also with external objects by using the tag <object>. Each fragment in a page is associated with a condition. These conditions are defined using the <if> tag, with one or two fragments inside <block> tags. When the conditional expression evaluates to *true* the user is shown the first fragment, otherwise the user is shown the second fragment. In case only one fragment is used, that fragment will only be shown if the Boolean expression is true [De Bra et al, 2003].

2.4.1 Architecture

AHA! uses Java Servlets to serve web pages either located in the local file system or in external http servers. The conceptual structure of the system consists of a combined domain and adaptation model (DM/AM), and adaptation rules. When a Servlet receives a request for a page, the DM/AM is used to select the most suitable page, which triggers adaptation rules. These rules result in updates to the UM and this new information is used to choose the conditional fragments and links that are to be included in the chosen web page. The servlet then serves this page and the user model is updated every time the user interacts with the application [De Bra et al, 2003].

The AHA! installation package is accompanied by instructions on how to integrate and use the system with the Apache Tomcat open source web server. All the information related to the DM/AM and the UM can be kept either in XML files or in a SQL database. The system's manager is responsible for making that choice. It's also the manager that has the responsibility of defining several installation parameters, such as installation directory, as well as creating accounts for authors [De Bra et al, 2003].

Various authoring tools have been developed for AHA!. The Concept Editor allows users low-level access to adaptation rules, whereas the Concept Graph serves the purpose of specifying the relationship between concepts. There's also a Form Editor, with which authors can create forms that let the user change some of the user model's attributes, and an Application Management Tool that allows authors to transfer files from their computers to the server and vice-versa [De Bra et al, 2003].

Figure 6 illustrates the global architecture of AHA!.

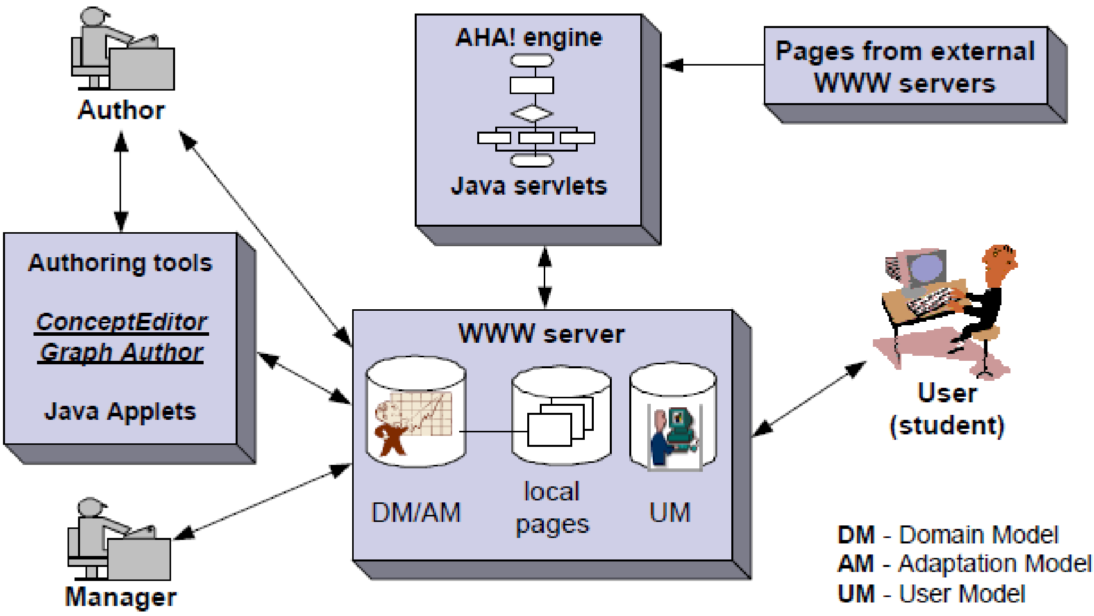

Figure 6 – AHA!'s architecture [De Bra et al, 2003].

2.4.2 The Domain/Adaptation Model and the User Model

Applications created under AHA! are made up of concepts. These concepts are associated with pages, fragments within pages or objects. Concepts can have various named attributes, and authors are the ones who establish which attributes a concept is associated with. AHA!

uses an overlay model, therefore all the attributes present in the DM/AM are present in the UM as well [De Bra et al, 2003].

AHA! works by receiving HTTP requests and sending XHTML documents via HTTP responses. Whenever a user interacts with an AHA! application, the system receives an HTTP request which may reference a concept or a page. Concepts can have resources associated with them, usually in the form of a page. If the request URL contains a page, the system searches for the corresponding concept in the DM [Hill and Carver Jr., 2000].

The User Model is updated with basis on the adaptation rules defined by the author. Once the concept included in the request has been identified, the rules linked to the corresponding access attribute are activated. Rules contain a condition and an action. The condition part of a rule consists of a Boolean expression that uses concept attributes. The action part assigns values or expressions to concept attributes. For example, "if page A is accessed, then the visited attribute of A is incremented by 1". It's possible to include a second action part, which will be activated if the condition evaluates to *false*. Since a rule's action might consist of an expression involving concept attributes, that rule can be used to activate another rule, which in turn might be used to activate yet another rule, and so on. An example of such a situation would be the case where a user visits the page of a particular painting in a museum, which causes an increase to the interest attribute of that painting's painter, which in turn might cause an increase to the interest attribute of the painter's style, etc. This behavior confers AHA! great power and flexibility. However, it can also be quite complex, making it hard for authors to predict the consequences of rule execution and therefore appropriately define all rules of an application. It may also happen that a sequence of rule actions results in an infinite loop. To deal with this problem and prevent run-time errors, AHA! limits the rule execution [De Bra et al, 2003][Hill and Carver Jr., 2000].

Updates to the UM occur before a web page is parsed and displayed. To better understand why this takes place, the prerequisite relationships between concepts should be considered. Using as example an educational application, it's desirable that a user learns about the concept of addition before moving on to the concept of subtraction. For this reason, users should only be given access to pages with information about subtraction after they've learned about addition. A page about addition might include a link to a page about subtraction, but that link should only appear if the user has already read the content about addition. If updates to the UM were performed after the page was displayed, the link to subtraction wouldn't be present in the page about addition. This would happen because when parsing the page, the information in the UM would indicate the user hadn't learned about addition yet. Updating the UM in this way might seem intuitive because only after seeing a page is the user able to read it. However, if the UM were indeed updated after the page about addition was parsed, the link to the subtraction page would only be shown if the user visited the addition page more than once. This is, of course, not a desirable behavior. On the contrary, by performing updates to the UM before the page is parsed, the link to subtraction will be included upon the user's first visit to the page, because when parsing the page it will already be indicated in the

UM that the user has read about addition. It seems adaptation should ideally happen while the user is reading a page, but more research into the subject is necessary [De Bra et al, 2006].

In AHA! a page's suitability can be determined in two ways. The first way involves the evaluation of a requirement expression, which is associated to every page or concept. If the expression is true the page is considered suitable. The suitability can also be determined by using it as an attribute and evaluating it in a Boolean expression. With this second method it's possible to determine the suitability of a page or concept using adaptation rules [De Bra et al, 2003].

The visited attribute is used to verify whether a page has been visited (or an object used) or not in AHA!. When using one of the authoring tools to create the adaptation rules, a rule that manages this attribute is created automatically. However, the visited attribute is a regular attribute of the UM, so it's also possible to use it in customized adaptation rules to achieve other behaviors [De Bra et al, 2003].

2.4.3 Adaptation

This section introduces the different types of adaptation performed by AHA!.

2.4.3.1 Forward and Backward Reasoning

Reasoning in AHA! can occur in two distinct manners: forward reasoning and backward reasoning. The way in which these two methods of reasoning work is best illustrated by using an example. For that purpose, the prerequisite relationship between addition and subtraction will once again be considered, but in this case it will also be assumed that information about subtraction should only be presented if the user's knowledge of addition has a value greater than or equal to 50% [De Bra et al, 2006].

With forward reasoning, the knowledge attribute of the concept addition is incremented when the UM is updated before parsing the page about addition. If the new value is at least 50%, an adaptation rule associated with the suitability attribute of subtraction will be triggered and that attribute will be changed to *true*. When parsing the page, and choosing whether or not to show the link to the page about subtraction, the system simply checks the link's suitability and renders it visible [De Bra et al, 2006].

With the exception of the access attribute, all other concept attributes in AHA! belong to one of two states: persistent or volatile. Attributes are also initialized to default values, which might consist of a value or an expression. Forward reasoning works by updating persistent attributes. Attributes defined as persistent are updated by adaptation rules. This is the case of the suitability attribute, which with forward reasoning is changed by the adaptation rule that was triggered by the update to the knowledge attribute [De Bra et al, 2006].

In the case of backward reasoning, no update is performed on the suitability attribute of addition, even if the knowledge value is 50% or higher. It's only when the page is parsed that the system verifies if the link is suitable by checking the knowledge value of addition. If the user has acquired the minimum knowledge required about addition, the system will display the link to the page about subtraction. Attributes that are updated through the evaluation of an expression are considered volatile. The attributes used in such an expression might be persistent, or volatile. If they're persistent the result of the expression is determined using their values. If they're volatile the expressions that define them are in turn evaluated and this process continues until a final value is reached. For this reason, backward reasoning performs updates to volatile attributes [De Bra et al, 2006].

The example used demonstrates that with forward reasoning the suitability of the link is determined while updating the UM and before the page is parsed, while with backward reasoning that process happens after updating the UM and during the parsing of the page [De Bra et al, 2006].

The advantage of forward reasoning is that it basically consists in verifying the value of an attribute. The downside is that it might be needlessly updated several times. With backward reasoning, on the other hand, the attribute is checked only when it's necessary, but since it involves evaluating a requirement expression it is slightly more time consuming than forward reasoning [De Bra et al, 2006].

AHA! gives the user (author) the freedom to combine these two methods of reasoning as he sees fit. Since this can result in infinite loops, AHA!'s authoring tools warn the author whenever there's a possibility that might happen [De Bra et al, 2006].

2.4.3.2 Adaptive Presentation

AHA! offers three ways of performing adaptive presentation: adaptive inclusion of fragments, adaptive inclusion of objects and adaptive changing of presentation style.

In AHA! the adaptive inclusion of fragments consists in embedding them into a page. If the suitability expression of the embedded fragment is true, the fragment becomes visible. This technique has the disadvantage of using the <if> tag, which results in pages that aren't standard XHTML [De Bra et al, 2003][De Bra et al, 2006]. The following (Code 1) is an example of how a fragment might be included in a page [De Bra et al, 2006]:

```
In Xanadu
 <if expr="myapp.xanadu.knowledge==0">
  <block>
      (a fully distributed hypertext system, developed by Ted Nelson
      at Brown University, from 1965 on)
  </block>
 </if>
```

```
there was only one protocol, ...
```

Code 1 – Example of fragment inclusion

As the example shows, there is a fragment of text embedded in the web page which is shown to the user if he has no knowledge of the concept “Xanadu”. It’s also possible to include another <block> of text, which would be shown if the expression evaluated to *false* (in this case that would happen if the knowledge value was different from zero). This technique can be used to make simple presentations of conditional content, such as prerequisite explanations that appear only in one place, or for presenting small sections of text that can alternatively be replaced by an image or video [De Bra et al, 2003][De Bra et al, 2006].

The inclusion of objects is better suited for more complex adaptive presentations than the inclusion of fragments. Conceptually this technique is similar to the previous one, but it’s quite different when put into practice. For objects to be included in a page they must be of the special type “aha/text”, and the tag <object>, which is part of the XHTML standard, must be used [De Bra et al, 2003][De Bra et al, 2006]. For example, the following line of code (Code 2) could be used to add an external object to a page [De Bra et al, 2006]:

```
<object name="myapp.conditionalobject" type="text/aha" />
```

Code 2 – Example of object inclusion.

Objects have identification names that represent concepts in the DM. In the example above the concept to which the object is associated is named “conditionalobject”. When an object is loaded, the access attribute of its concept in the DM triggers adaptation rules and causes the UM to be updated [De Bra et al, 2003]. The showability attribute of the concept “conditionalobject” is then used to choose the resource file that will be included in the page. An important difference between this technique and the previous one is that the conditional inclusion of objects causes updates to the UM while the page is being constructed. For this reason, after an object has been included the adaptation of the remainder of the page will be affected by the changes that have been made to the UM [De Bra et al, 2003][De Bra et al, 2006].

The dialog box in figure 7 shows which resource should be chosen when the object named “object“ is added to a page. Since updates to the UM occur before the page is parsed, the visited attribute of a page changes to 1 when it’s first visited by a user. For that reason, the default resource, which is shown when the attribute visited has the value 0, is never included in the page. Upon the user’s first visit, he’ll be shown “object1”. On his second visit he’ll see “object2”, and on subsequent visits the included object will be “object3”. However, the way in which resources are associated with objects might have unintended consequences. For example, if a page contains the same object three times, when a user sees the page for the first time he’ll see “object1”, “object2” and “object3”. This happens because with the adaptive inclusion of objects the UM is updated while the page is being constructed. When “object1” is added to the page, the value of the attribute visited is changed to 2. As a result, when the object is added to the page a second time the resource chosen will be “object2”. The visited attribute is changed once again when “object2” is included in the page, so the next addition to the page will be “object3”. If the user decides to visit the page a second time he’ll see the same object three times. This happens because by then the visited attribute will have a value greater than 3 and according to the definitions shown in figure 7 the only resource that must be shown in that case is “object3” [De Bra et al, 2003][De Bra et al, 2006].

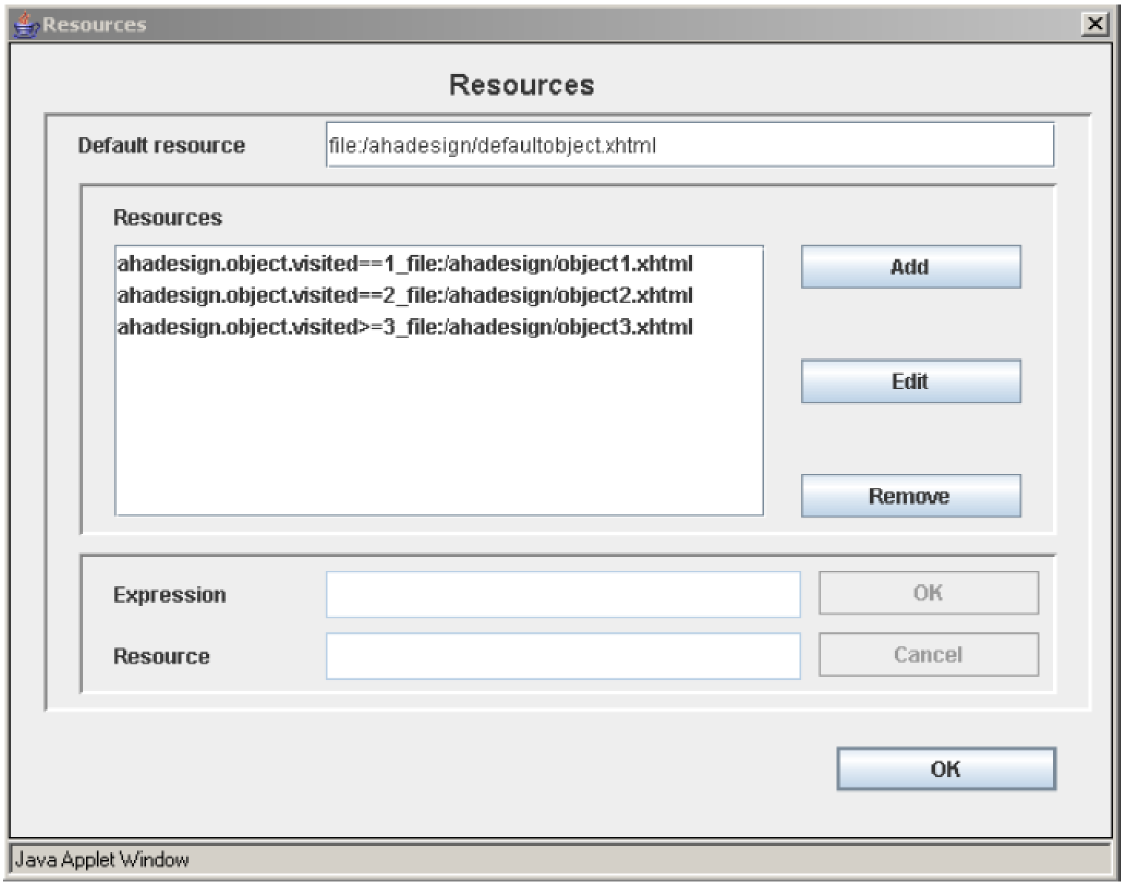

Figure 7 – Definition of the resources associated with an object.

Adaptive changing of presentation style consists in highlighting or downplaying a particular fragment by changing the way it is presented. The system uses the visibility attribute to achieve this purpose. It appends the letter “c” to the value of the visibility attribute and selects the corresponding class in a cascading style sheet. For example, if the visibility attribute of a fragment has the value 30 it will be presented using the style c30, which in the default style sheet consists in showing the background colored in light gray and the text colored in a darker shade of gray. In the default style sheet the class c30 serves the purpose of dimming the text of a particular fragment. Style c40 represents normal text presentation, while style c60 presents text in italic and style c70 shows underlined text. Although AHA! comes with a default style sheet, each AHA! application can have its own style sheet and use it to present classes c20 to c100 in the desired manner [De Bra et al, 2003][De Bra et al, 2006].

2.4.3.3 Stable Presentation

The standard behavior of an adaptive system causes pages to be adapted to the UM each time they're accessed by the user. Such a behavior isn't necessarily desirable because it means pages aren't stable. Due to constant UM updates, a page consulted a second time might look very different from the first time the user saw it [De Bra et al, 2003][De Bra et al, 2006].

To deal with this issue AHA! allows authors to choose the stability of concepts. It's possible to define the stability of each page and object of an application, but not the stability of fragments included using the <if> tag. Authors can choose from four distinct stability states:

- no stability - This refers to the standard behavior of AHA! where a page (or object) is adapted to the current state of the UM every time it is visited [De Bra et al, 2003][De Bra et al, 2006].
- always stable - With this option a page (or object) will be adapted to the UM's state the first time it's visited by a user and remain that way in subsequent visits. The color of links, however, will still suffer changes based on their visited status as that is the normal behavior users are used to seeing in a web page [De Bra et al, 2003][De Bra et al, 2006].
- session stability – A page (or object) will be adapted to the UM's state the first time it's visited by a user and remain that way for the duration of the current session [De Bra et al, 2003][De Bra et al, 2006].
- conditional stability – This option allows the author to associate an expression to the page's stability. The page's presentation will remain stable while the expression is true, and be adapted to the UM's current state when the expression becomes false [De Bra et al, 2003][De Bra et al, 2006].

Although it's probably not a very common situation, it's possible to include the same object several times in the same page. If AHA! only allowed object stability to be defined, the stability of the whole page would be compromised. For that reason, AHA! gives the author the possibility of defining both object stability and page stability [De Bra et al, 2003][De Bra et al, 2006].

2.4.3.4 Adaptive Navigation Support

AHA! provides adaptive navigation support in three different ways*:* generation of views (lists of links), adaptive link annotation and adaptive link destinations.

As the name suggests, generation of views consists in adaptively creating lists of links for different views. AHA! allows authors to define the presentation style of their applications and provides support for the creation of layouts. A layout can have one or more viewgroups, which are made up of views. Authors may create views of their own, but can also use the one included in the AHA! distribution. The *StaticTreeView* is one such view and is used, among others, to display adaptively created lists of links. The links in these lists may be sorted adaptively according to a user's characteristics. The way in which a list of links is presented is also determined by the hierarchy of concepts and the position in that hierarchy of the page being viewed at the moment [De Bra et al, 2003][De Bra et al, 2006].

Link annotation in AHA! is also adaptive. The way links are presented can be different for each view. A typical way of adapting the presentation of a link consists in using the suitability and visited attributes of the page the link connects to. The link will be shown using a "good" color if the page is suitable but hasn't been visited yet and a neutral color if it has already been visited. Links that aren't suitable will appear with a "bad" color. There can be as many of these abstract colors as the author desires [De Bra et al, 2003][De Bra et al, 2006]. The following code (Code 3) illustrates how the link anchor adaptation can be specified [De Bra et al, 2006]:

```
<linkanno>
 <linkclass expr="!suitability">Bad</linkclass>
 <linkclass expr="suitability &amp;&amp; visited &gt; 0">
   Neutral</linkclass>
 <linkclass expr="suitability &amp;&amp; visited == 0">Good</linkclass>
</linkanno>
```

Code 3 – Definition of the link anchor adaptation.

The color scheme for a link is defined in the file "ConceptTypeConfig.xml" using the tag <linkanno>. The author can choose the colors he wants for the abstract color categories shown above, but the colors usually used are blue for *good*, purple for *neutral* and black for *bad*. It's also possible for users to change the color scheme of links if the application was created with that option enabled. When a link is deemed "bad" and appears in black it looks just like normal text and is thus hidden from the user, which corresponds to the link hiding technique of adaptive navigation support. Adaptively placing icons next to a link's anchor is another common way of performing link annotation. A concept's suitability and visited attributes may be used to choose an icon that reflects whether the link is suitable and has been visited or not, but other attributes, such as the knowledge attribute, may be used as well [De Bra et al, 2003][De Bra et al, 2006]. The following code (Code 4) demonstrates how to define the adaptive icon generation [De Bra et al, 2006]:

```
<iconanno>
 <icon expr="knowledge&lt;40&amp;&amp;visited&gt;0"
   view="StaticTreeView">icons/SmallCheckM.gif</icon>
 <icon expr="knowledge&gt;39&amp;&amp;knowledge&lt;75"
   view="StaticTreeView">icons/MedCheckM.gif</icon>
 <icon expr="knowledge&gt;74"
   view="StaticTreeView">icons/BigCheckM.gif</icon>
</iconanno>
```

Code 4 – Definition of an adaptive icon.

To place an icon next to a link the author must define the conditions that result in the presentation of the icon, as well as indicate the URL of the image file to be used and where next to the link the icon should be placed (in front or behind). In the code example above (Code 4), there is no indication of where the icon should be placed, so by default it is placed behind the link. It's possible for more than one icon to be put next to the same link. All the icons whose expressions evaluate to *true* will be placed in front or behind it. These are two common ways of adapting the presentation of links, but authors are also given the freedom to fully customize it [De Bra et al, 2003][De Bra et al, 2006].

AHA! offers a third form of adaptive navigation support that was specifically designed by its authors for this purpose. Adaptive link destinations works in a manner akin to forward and backward reasoning. With forward reasoning a link's destination is chosen before placing the link in a page. Conditional fragments are used to include in the page links that have the same anchor but lead to different pages. Which of these is shown is determined when building the page, so the link's destination is also decided in this moment. On the contrary, when using backward reasoning the destination is chosen only after the page has been constructed, at the moment the link is followed. This can be achieved by using the showability attribute of the concept associated with the link to determine which page the link points to, but other attributes may be used as well [De Bra et al, 2003][De Bra et al, 2006].

Figure 8 shows how to associate a resource with a link on the basis of a specific conditional expression. When the visited attribute of the concept "authoring tools" has a value greater than zero the link's destination is the file "linkdestination2.xhtml", when the value is smaller than one the link's destination changes to file "linkdestination1.xhtml". The default resource associated with the link will never be used since all possible values of the visited attribute are covered by the conditional expressions [De Bra et al, 2003][De Bra et al, 2006].

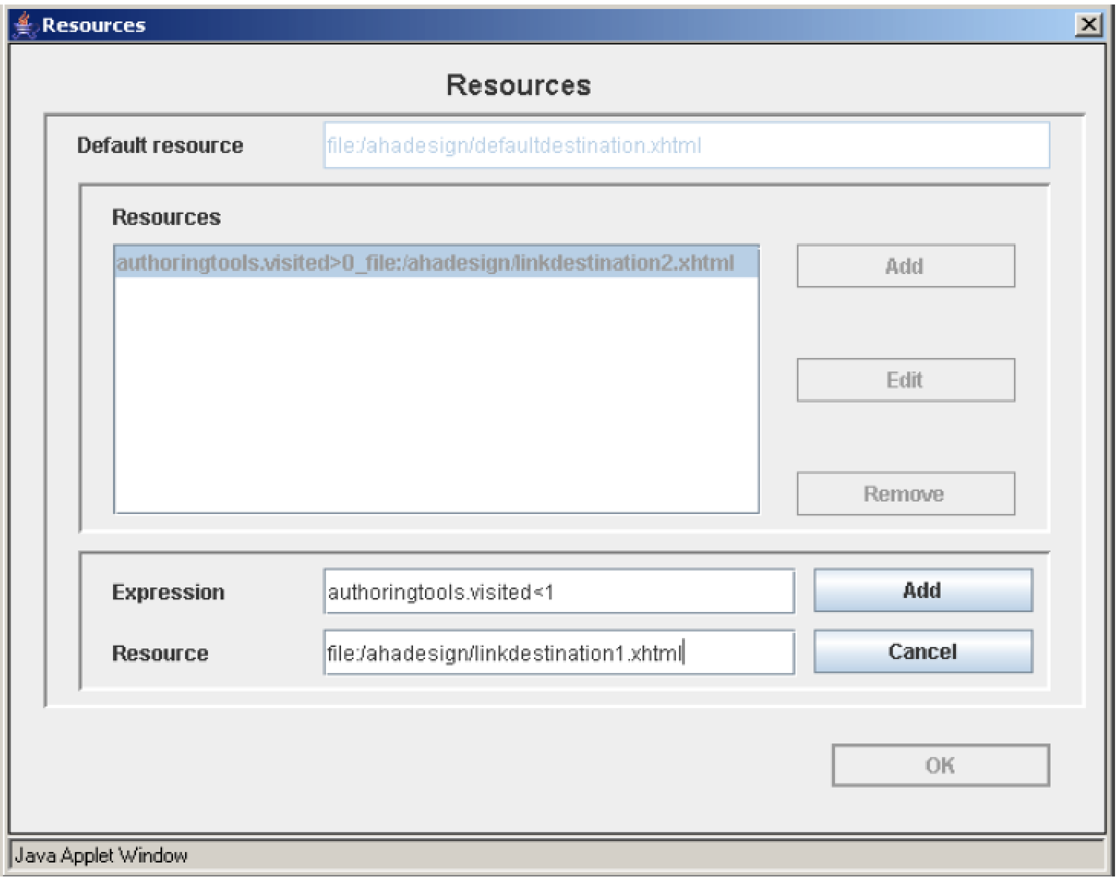

Figure 8 – Association between a resource and a link [De Bra et al, 2006].

2.4.4 Authoring Tools

Applications created in AHA! comprise a conceptual structure and information content. Information content must be in the form of XML or (X) HTML and can therefore be produced using any tool that serves that purpose. AHA! has handlers for processing the different content types. The handlers that process HTML and XHTML files allow the author to combine the standard (X) HTML tags with AHA!'s specific tags (the <if> tag, for example). There's also a handler for processing XML and SMIL files, which differs from the (X) HTML handler in that it doesn't manipulate the page's headers. AHA! accepts any valid XML format [De Bra et al, 2006].

The complexity of the DM/AM has increased greatly since version 1.0. Each concept can have as many attributes as desired, and each attribute can be associated with rules that update as many attributes as desired, for as many concepts as desired. For this reason, creating applications with AHA! can no longer be done by directly creating the XML and HTML files, and must be achieved through the use of authoring tools that have been developed specifically for that purpose [De Bra et al, 2003][De Bra et al, 2006].

The Concept Editor is one of the authoring tools included in AHA!. It uses Java Applets to provide the author with a graphical means of defining concepts and adaptation rules. The Concept Editor is a low-level tool that gives the author a lot of control over an application. The author defines the template that will be used by the editor to relate attributes and rules to each concept. The author must also specify all the adaptation rules between concepts. Since it's common for an application to have relationships that are repeated several times, such as the knowledge propagation from page to section to chapter, using the Concept Editor involves a great deal of repetitive work [De Bra et al, 2003].

To deal with the problem of monotonous work inherent to the use of the Concept Editor, the creators of AHA! developed a tool that uses high-level concept relationships: the Graph Author. This tool is also a Java Applet-based tool that allows the author to associate attributes and adaptation rules to each concept, but it facilitates this process by giving the author the ability of creating a graph to do so [De Bra et al, 2003].

2.4.4.1 Graph Author

Unlike the Concept Editor, this tool uses high-level concept relationships. It provides the author with templates for different concept relationships and allows him to define the concept hierarchy by simply drawing a graph structure. Using the templates provided means the author doesn't need to know much about the adaptation mechanism, since the Graph Author automatically generates the necessary low-level concept structures and adaptation rules. The author needs to work with conditional expressions only when defining the conditional inclusion of objects and adaptive link destinations. However, the author can also create new templates and for that he needs to be aware of how adaptation rules are built, as well as how they are processed by AHA!'s adaptation engine [De Bra et al, 2003][De Bra et al, 2006].

Figure 9 shows a concept hierarchy created with the Graph Author. This hierarchy determines how the user model is updated. For example, if viewing a page about a specific concept results in an increase of the value of the *knowledge* attribute of that concept, it will also cause the value of the *knowledge* attribute of its parent concept to increase, as long as such a behavior has been specified in the concept hierarchy (by using the appropriate templates). The author can create a graph such as the one shown in the figure, by dragging concepts into the graph pane and connecting them with concept relationships. In the graph below there are prerequisite relationships between the concepts that form the hierarchy, but there are also relationships that indicate different presentation styles for different fragments of text [De Bra et al, 2006].

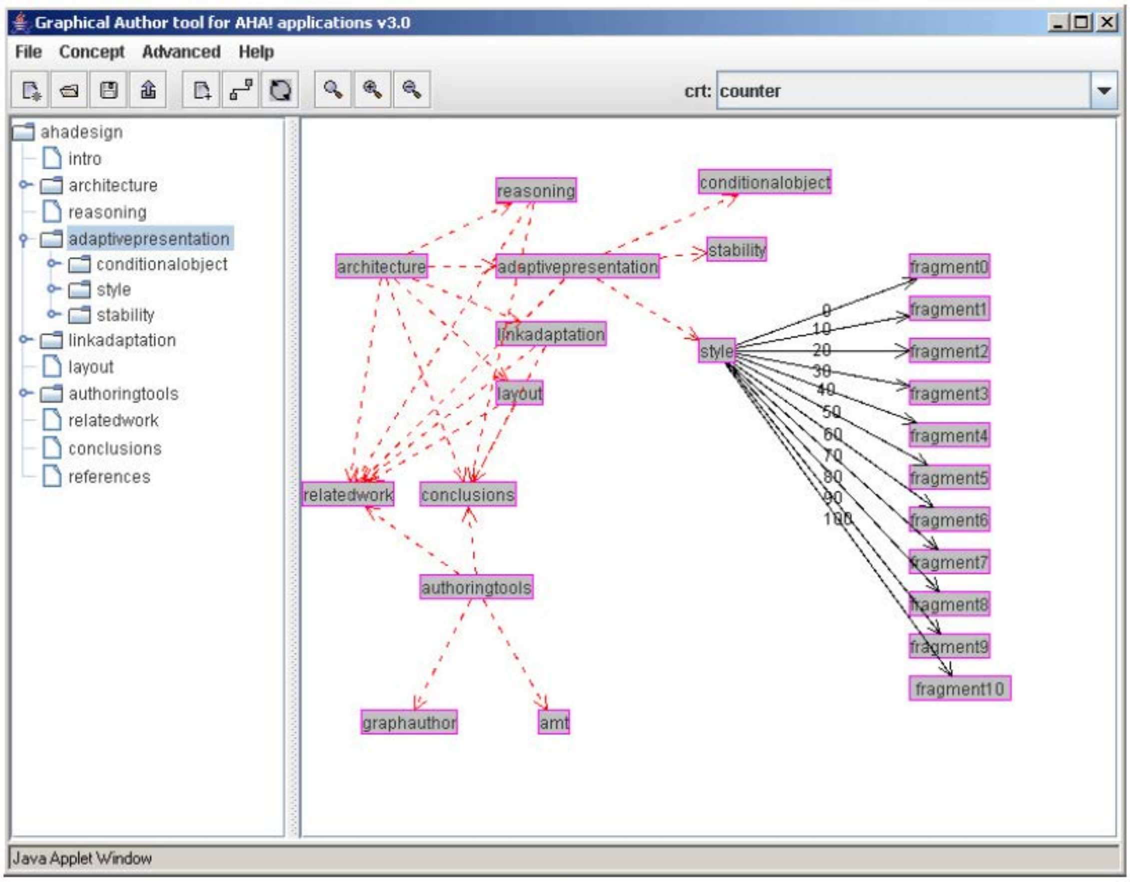

Figure 9 – The Graph Author [De Bra et al, 2006].

While the Graph Author facilitates the graphical creation of the concept structure, the author still has to do all the work related with designing it. It's up to the author to define how concepts are related to each other, which text fragments and objects to use where, and when, and which destinations links lead to, under which circumstances [De Bra et al, 2006].

3 Mathematics Collaborative Learning Platform – PCMAT

This chapter presents the Mathematics Collaborative Learning Platform. A short introduction of the platform is given in section 3.1, followed by a presentation of its architecture in section 3.2. Sections 3.3 and 3.4 describe PCMAT's student model and domain model, respectively, and section 3.5 addresses the creation of learning objects.

3.1 Introduction

PCMAT takes into account the constructivist learning theory. The system assesses the user's previous knowledge and provides a path into the subject based on it. It also presents the user with learning objects, such as content and activities, which are adapted to the student's characteristics and performance. In addition, with the purpose of consolidating the user's knowledge, the system is able to make permanent automatic feedback and support, through instructional methodologies and educational activities explored in a constructivist manner.

PCMAT is based on AHA!. For that reason, the adaptation model of the platform can be divided into two, intrinsically connected, parts: one that is dependent on AHA's adaptation engine and another that was developed under the scope of this thesis to address our platform's specific needs.

3.2 Architecture

PCMAT's architecture consists of three main components: student model, domain model and adaptation model.

The student model keeps a record of the student's knowledge and preferences, as estimated by the system. This model is platform-independent and can therefore be used in other AHS.

The domain model consists of a graph of concepts that serves as a structure for the representation of the domain.

The adaptation model defines the adaptation rules and the interaction mechanisms between the user and the system. The information obtained is used to infer some of the user's characteristics and update the student model.

PCMAT was developed using a component-based approach that simplifies the substitution and addition of new components. The system stands out from AHA!, on which it is based, as well as from other AEHS in the following ways:

- The definition of the Student Model takes into account the user's learning preferences;
- The adaptation rules outlined in the Adaptation Model are based on the constructivist theory, in the sense that the system assesses a student's current knowledge and uses it with the student's learning preferences, as well as other characteristics, to adapt the activities and content the student sees.

The AHA! runtime environment consists of Java Servlets and uses a XML database representation. We have chosen to develop the PCMAT adaptation engine using the same technologies. The system also works with freely available tools like the Tomcat Web server and the MySQL database management system.

PCMAT has a frontend/backend structure. The frontend was implemented using XHTML, JavaScript and CSS. The backend is composed by several modules developed using Java classes and Java Servlets.

The module responsible for recommending learning objects, the module that generates activities adapted to a student's current knowledge and preferences, and the module that evaluates the answers given to those activities are part of the backend. These modules, consisting of Java Servlets and auxiliary Java classes, are integrated in AHA!'s backend and cooperate with its Servlets seamlessly. PCMAT's backend also includes a module for the search and retrieval of learning objects. This module consists only of Java classes and isn't integrated into AHA!, but communicates with and provides its services to the module that recommends learning objects. All of these modules connect to the database through the data access layer.

The system's frontend is made up of Web pages that were created using XHTML, CSS, and JavaScript. These pages may be composed of several fragments in which are inserted learning objects that are adapted to each student's personal characteristics, learning style and knowledge. The content inserted into these fragments is generated at run time by the recommendation module. A similar process is used when presenting the student with activities used for progressive assessment.

PCMAT also includes an authoring tool that helps course authors create activities/tasks and place them in a course in a simple manner that respects the system's information structure. The tool's frontend was developed using XHTML, CSS and JavaScript, and the backend was developed in Java. The information inputted by the user is transferred from the client to the server, where it is process by the business logic layer and stored in a MySQL database.

Figure 10 shows a detailed representation of the system's architecture.

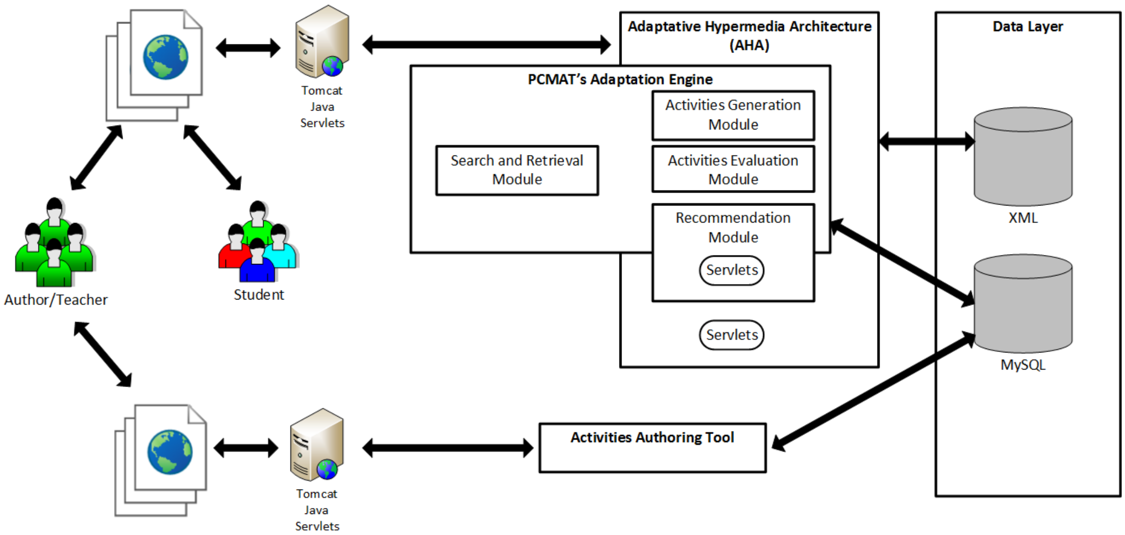

Figure 10 - Detailed representation of PCMAT's architecture.

3.3 Student Model Implementation

PCMAT's User Model was implemented by combining the overlay method, for the knowledge representation of the student, with the use of stereotypes [Martins et al, 2013].

The user modeling process begins with the identification of the stereotype better suited to a given user. This is achieved by using questionnaires and learning styles (Figure 11). The reliability of each questionnaire was determined by computing Cronbach's alpha coefficient [Woodward and Chambers, 1983] using IBM's SPSS [SPSS, 2015]. The alpha coefficient obtained for the learning styles questionnaire was 0.91. This value is a good indicator that the internal consistency of the questionnaire is very good.

Questionário de Escolha Múltipla

Para cada pergunta escolhe a resposta que melhor se adequa ao teu comportamento.

1. Se no decorrer duma viagem não tivesses a certeza onde te encontravas, qual seria a tua primeira opção?

Procuravas um mapa.	Perguntavas a uma pessoa que fosse a passar.	Seguias por onde achasse melhor.

2. Se tivesses que ensinar algo a um amigo, como o farias?

Escrevias a matéria para ele ler.	Explicavas oralmente.	Demonstravas como se fazia.

3. Normalmente, quando explicas algo tens tendência para dizer:

Vê como eu faço.	Ouve a minha explicação.	Tens que experimentar.

4. Quando estás a aprender uma matéria nova, preferes:

Ver o que o professor faz.	Perguntar ao professor como se faz.	Experimentares fazer por ti mesmo.

5. Para te concentrares na resolução dum problema, costumas:

Focar a tua atenção no enunciado.	Imaginar as soluções possíveis.	Mover-te dum lado para o outro ou "brincar" com um lápis ou uma esferográfica.

6. Quando estudas para um teste ou exame:

Escreves diversas notas e esquemas.	Falas sobre a matéria sozinho em voz alta ou com outros colegas.	Revês a matéria mentalmente.

7. Quando explicas algo a alguém: [semelhante a 2]

Mostras o que pretendes explicar.	Explicas de diferentes maneiras até a outra pessoa entender.	Incentivas a outra pessoa a experimentar e vais explicando.

8. Gostas mais de:

Ver filmes, fotografias, obras de arte ou observar as pessoas.	Ouvir música, rádio ou conversar com os amigos.	Participar em atividades desportivas, comer ou dançar.

9. Passas a maior parte do tempo livre a: [semelhante a 8]

Ver televisão.	Falar com os amigos.	Praticar desporto ou outras atividades.

10. Quando estás ansioso, como por exemplo na véspera dum exame:

Imaginas o pior dos cenários.	Pensas sobre o que mais te preocupa.	Não consegues ficar quieto.

Clica neste botão [Iniciar PCMAT] para efetuares o registo, ou neste botão [Reset] para limpares as respostas que deste e começares de novo.

Figure 11 - Questionnaire used to determine users' learning styles.

The student's characteristics are defined taking into account the domain model and the application's constructivist approach. Table 2 presents a generic student profile used by

PCMAT. Not all of the characteristics listed are used in the current version of the platform, but their application is planned for future iterations.

Table 2 - Characteristics used in the student model [Martins et al, 2013].

Model	Profile	Characteristics
Domain Independent Data	Generic Profile	- Personal information - Demographic data - Academics background - Qualifications - Knowledge (background knowledge) - Disabilities: visual or others - Inheritance of characteristics
	Psychological profile	- Learning styles - Cognitive capacities - Personality traits - Inheritance of characteristics
Domain Dependent Data		- Objectives - Planning / Plan - Complete description of the navigation - Acquired knowledge - Evaluation results - Interests

The tools employed to collect the necessary data were:

For the domain independent data:

- Questionnaires
- Learning Styles questionnaires
- Psychometric tests

For the domain dependent data:

- Questionnaires
- Tests

In PCMAT, the definition of learning preferences is based on a combination of David Kolb's Model [Kolb, 1984] with Neil Fleming's VAK model [Fleming and Mills, 1992].

Kolb's Learning Style Inventory was chosen because it's simple to understand, and allows the implementation of a very flexible solution. The VAK model was chosen due to its simplicity, making it easy to assess a person's dominant learning style. It establishes three learning preference categories: visual, auditory and kinesthetic. This model is used to categorize PCMAT's contents and activities.

Mapping between David Kolb's and Neil Fleming's models facilitates the system's choice of the most suitable content and activities for a given student. Table 3 illustrates how this mapping is performed. A strategy similar to this one could not be found in the literature.

Table 3 - Mapping of learning styles.

Kolb's Learning Styles	VAK Model
diverging	visual
assimilating	auditory
converging	kinesthetic
accommodating	kinesthetic

The student model is updated by means of the adaptation model, specifically, by using activities to assess the student's performance and by monitoring his interaction with the system (Figure 12).

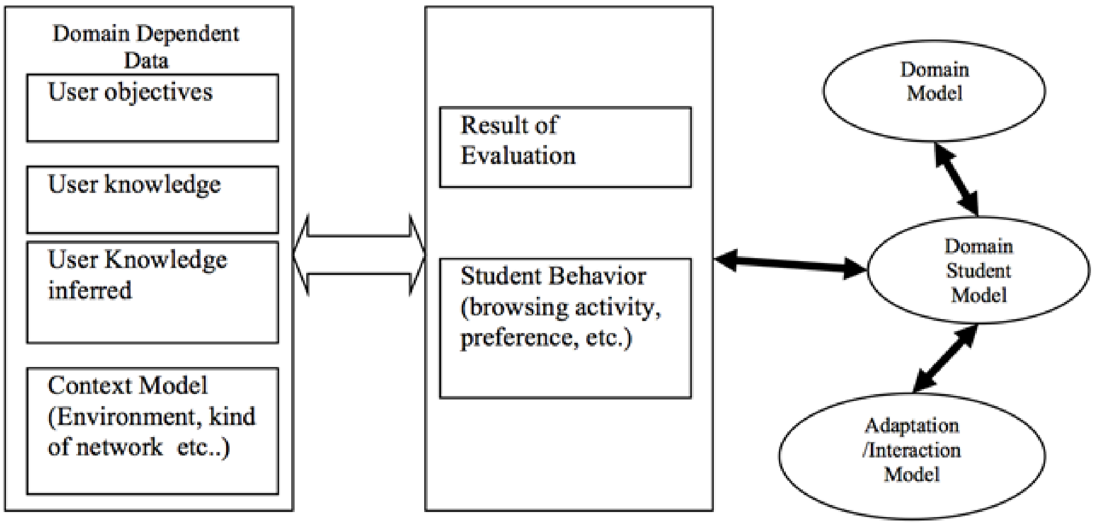

Figure 12 - PCMAT's Domain dependent data architecture [Martins et al, 2013].

Each student's profile is kept in an XML file. This XML file contains the information related to the domain dependent data, as well as the domain independent data. The structure of the data and its type are validated by PCMAT's student model schema (Code 5).

```
<xsd:element name=" Student_Model ">
 <xsd:complexType>
 <xsd:sequence minOccurs="1" maxOccurs="1">
 <!--definition of data related with DDD and DID -->
 <xsd:element name="Domain_Independent_Data"
        type="TDomain_Independent_Data"/>
  <xsd:element name="Domain_Dependent_Data"
        type="TDomain_Dependent_Data" />
```

```
  </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
 <xsd:complexType name=" TDomain_Independent_Data ">
  <xsd:sequence>
  <xsd:element name="Generic_Profile"
        type="TGeneric_Profile" />
  <xsd:element name="Cognitive_Profile"
        type="TCognitive_Profile" />
  </ xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name=" TGeneric   Profile ">
  <xsd:sequence minOccurs="1" maxOccurs="1">
  <xsd:element name="Personal_Information"
        type="TPersonal_Information" />
  <xsd:element name="Academic_Background" type="TAcademic" />
  <xsd:element name="Demographic_data"
        type="TDemographic_data" />
  <xsd:element name="Background_Knowledge"
        type="TBackground_Knowledge" />
  </xsd:sequence>
 </ xsd:complexType>
 <xsd:complexType name=" TDomain_Dependent_Data ">
  <xsd:sequence>
  <xsd:element name="Domain_Knowledge"
        type="TBackground_Knowledge" />
  <xsd:element name="Task made" type="TTask_made" />
  ...
 ...
  </xsd:sequence>
 </xsd:complexType>
 ...
```

Code 5 - Student model schema excerpt.

3.4 Domain Model Implementation

Mathematics was the topic chosen as PCMAT’s domain - more specifically, the concepts related to the subject of Direct Proportionality. This choice was made with the assistance of the Mathematics teachers that were involved in the development and testing of this project.

There were two main reasons for the selection of this domain:

- According to the OECD PISA 2012 [OECD, 2014] study, Portugal is still below the OECD average in mathematics performance. Portugal's lower rank is 36 and upper rank is 26 out of 65 countries featured in the study.
- Direct proportionality is a topic covered in several school years, which guarantees a more diverse target audience.

The domain model consists of a semantic network of domain concepts. Its main purpose is providing a representation structure for the user’s domain knowledge. With the intent of clarifying the representation and visualization of the domain model, the textual form of the concepts used in PCMAT was converted into an alphanumeric code. For example, the concept *equivalent ratios* was encoded as *A1*.

PCMAT's domain model is represented by a graph named GPCMAT (Figure 13). It consists in a directed acyclic graph, whose edges define the prerequisites of each concept.

GPCMAT is then, the graph that represents the prerequisite relationships, where:

V: is the set of vertices (or nodes);

E: is the set of ordered pairs of vertices, called edges.

V = $\{A, A1, A2, A3, A4, B, B1, B2, B3, B4, B5\}$

$$E = \left\{ \begin{array}{c} (A, A1), (A, A2), (A, A3), (A, A4), \\ (A1, A2), (A2, A3), (A3, A4), (A, B) \quad , \\ (B, B1), (B, B2), (B, B3), (B, B4), (B, B5), \\ (B1, B2), (B2, B3), (B3, B4), (B4, B5) \end{array} \right\}$$

The following is an example of the structure of the XML file that represents the graph (Code 6).

```xml
<concept_hierarchies>
 <concept_relation>
        <concept_name>A</ concept_name>
        <hierarchy>
              <firstchild>A1</ firstchild>
              <nextsibling>A2</ nextsibling>
              <!-- root element of the concept graph !-->
              <parent>proporcionalidade</ parent>
        </hierarchy>
        <children>
              <concept_name>A1</ concept_name>
              <concept_name>A2</ concept_name>
              <concept_name>A3</ concept_name>
              <concept_name>A4</ concept_name>
        </children>
        ...
 </concept_relation>
 <concept_relation>
        <concept_name>A1</ concept_name>
        <hierarchy>
              <firstchild></ firstchild>
              <nextsibling> A2</ nextsibling>
              <parent>A</ parent>
        </hierarchy>
        <children></ children>
 </concept_relation>
</ concept_hierarchies>
```

Code 6 - Excerpt of the XML representation of the GPCMAT concept graph.

This XML file describes the relationships between concepts, and represents the graph GPCMAT. Each concept is defined by a name, and a hierarchy. This hierarchy includes two elements:

- Origin vertices (concept_parent), which stand for prerequisite concepts;

- Destination vertices (concept_children), or vertices of which the current vertex is a prerequisite.

The graph GPCMAT includes the paths a student might take if he successfully completes the required tasks. It doesn't, however, include the paths the student can take in case of lack of success.

To address this need, a different directed acyclic graph, named GI, was specified. The edges of this graph define the relationships between concepts in case of lack of learning success (Figure 13). In this way, if a student performs poorly on an activity, the graph's edges point towards the concepts to which the student should be redirected.

The GI graph is defined as such:

V_1: is the set of vertices;

E_1: is the set of ordered pairs of vertices, or edges.

V1 = $\{A, A1, A2, A3, A4, B, B1, B2, B3, B4, B5\}$

$$E1 = \left\{ \begin{array}{c} (A1, A), (A2, A), (A3, A), \\ (A4, A), (A4, A1), (B, A), \\ (B1, B), (B2, B), (B3, B), \\ (B3, B1), (B4, B), \\ (B4, B1), (B5, B) \end{array} \right\}$$

Figure 13 illustrates the GI graph, as well as the GPCMAT graph.

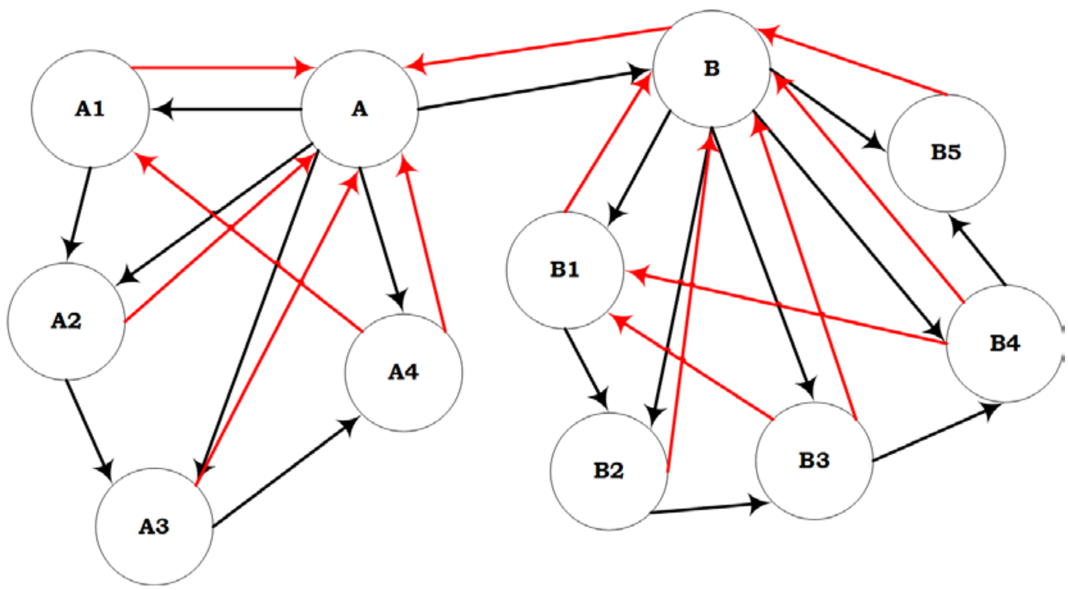

Figure 13 – The graphs GPCMAT (black edges) and GI (red edges).

3.5 Learning Objects

One of the main issues with which authors are faced when using electronic educational technology is the creation of learning objects (LO) that are in accordance with the educational purposes, and are pertinent to the student's needs. To address that need PCMAT was equipped with the tools necessary for the creation of learning objects that comply with an adopted standard. Learning objects that are in accordance with an established standard can be reused by different learning platforms and shared across repositories of learning objects.

Learning objects may have several different names such as resources, knowledge objects, instructional components, pedagogical documents, educational software components and online learning materials [Wiley, 2002]. There are also various definitions for learning objects. The Institute of Electrical and Electronics Engineers (IEEE) defines learning objects as "any entity, digital or non-digital, that may be used for learning, education or training" [IEEE, 2002]. According to David Wiley [Wiley, 2002], a learning object is "any digital resource that can be reused to support learning". It's this more restricted definition that best describes the learning objects used by PCMAT and the manner in which they are used.

The creation and usage of learning objects in PCMAT is based on the National Information Standards Organization's (NISO) definition of metadata. According to this association, metadata is "structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource" [NISO, 2004]. The concept of "structured information" implies the need for a metadata scheme that can be used to identify the characteristics of a given type of information resource. In this sense, a metadata scheme has both semantic meaning, because it defines each element in the scheme, and content, because each of those elements is given a value. There are currently two main metadata standards for the description of learning objects: the IEEE Learning Object Metadata (LOM) and the Dublin Core Metadata Element Set (DCMES).

3.5.1 A metadata standard for PCMAT

PCMAT allows both authors and teachers to manually retrieve learning objects. The platform itself also needs to retrieve learning objects and know which ones to present to which students. To do this, each learning object must be associated with a metadata record containing the following information:

- Descriptive metadata – Information about the creator of the learning object, as well as information necessary for the search and retrieval of the learning object, such a title, description and keywords;
- Administrative metadata – Details concerning when and how the learning object was created, technical characteristics of the learning object, and information about intellectual property rights;
- Educational metadata – Information regarding the age group of the intended audience, difficulty level and interaction level of the object, etc. This data must

facilitate the retrieval of a learning object compatible with a student's knowledge and learning style.

In order for the metadata standard adopted for PCMAT to be congruous with the purposes of the platform, it needed to be extensible, thus allowing the inclusion of new elements not previously described in the standard. It also had to allow a metadata instance to be represented in an XML file and, as such, supply the corresponding XML Schema.

Taking these requirements into account, the team working on PCMAT decided to choose one of the two metadata standards mentioned above, instead of creating a new metadata standard specifically for the platform. Given the different characteristics of the IEEE LOM and the DCMES, the IEEE LOM was chosen as the metadata standard for PCMAT because it defines a vast number of metadata elements and satisfies all the necessary requirements.

3.5.2 IEEE Learning Object Metadata

The IEEE LOM standard includes the following documents:

- IEEE Std 1484.12.1-2002 [IEEE, 2002] – this document describes the conceptual model underlying the structure of a metadata instance for any learning object.
- IEEE Std 1484.12.3-2005 [IEEE, 2005] – this document defines the XML structure and the restrictions applicable to XML1.1 documents used to represent the metadata instances of learning objects.

This standard establishes a structured set of seventy six elements that describe a wide range of features that are considered significant for the description of learning objects. These elements are grouped in the following categories:

- General – non-specific data used to describe the learning object, such as unique identifier, title, description, keywords, etc.
- Life cycle – details about the creation of the learning object.
- Meta-metadata – information concerning the metadata document, not the learning object it describes.
- Technical – data describing the features and technical requisites of the learning object.
- Educational – information describing the pedagogical and educational features of the learning object.
- Rights - information about intellectual property rights.
- Relation – Details on how the learning object being described is related to other learning objects.
- Annotation – observations regarding the usage of the learning object
- Classification – description of the learning object according to different classification systems.

Every element in every category is optional. Additionally, certain elements in each category may be repeated. The standard also states that "a LOM instance that contains no value for any of the LOM data elements is a conforming instance" [IEEE, 2002]. The following XML document is in accordance with this statement (Code 7):

```
<lom xmlns="http://ltsc.ieee.org/xsd/LOM"/>
```

Code 7 – example of a conforming LOM instance.

However, if the IEEE Lom standard is used to describe a learning object, the elements used must conform to the structure defined in the schema, and their value must be in agreement with the datatypes and vocabularies described in the schema. It's also possible to include elements and attributes not described in the standard, as long as these are included in a distinct namespace.

Owing to the fact that usage of all the elements described in the standard isn't compulsory and that the standard is extensible, a community of users has the freedom to create its own specific metadata structure, hence developing what is known as an *application profile*.

The IEEE LOM has been adopted by several repositories, such as: ARIADNE, SMETE, Learning Matrix, iLumina, MERLOT, HEAL, CAREO, LearnAlberta Online Curriculum Repository, and Lydia Inc [Couto et al, 2011].

3.5.3 Authoring Tool for the creation of Assessment Learning Objects

The PCMAT platform has an authoring tool that authors can use to create learning objects designed for assessment, also named activities or tasks. The frontend of the application was developed using XHTML, CSS and JavaScript, and the backend was developed in Java.

All activities have to be related to at least one concept and at most five concepts. Moreover, each activity has to be classified according to its compatibility with a learning preference.

The body of an activity may be directly inserted by the user, but it may also be introduced by uploading a text file. A resource, such as an image file, may be added as well. The following figure shows the user interface used for the creation of activities (Figure 14):

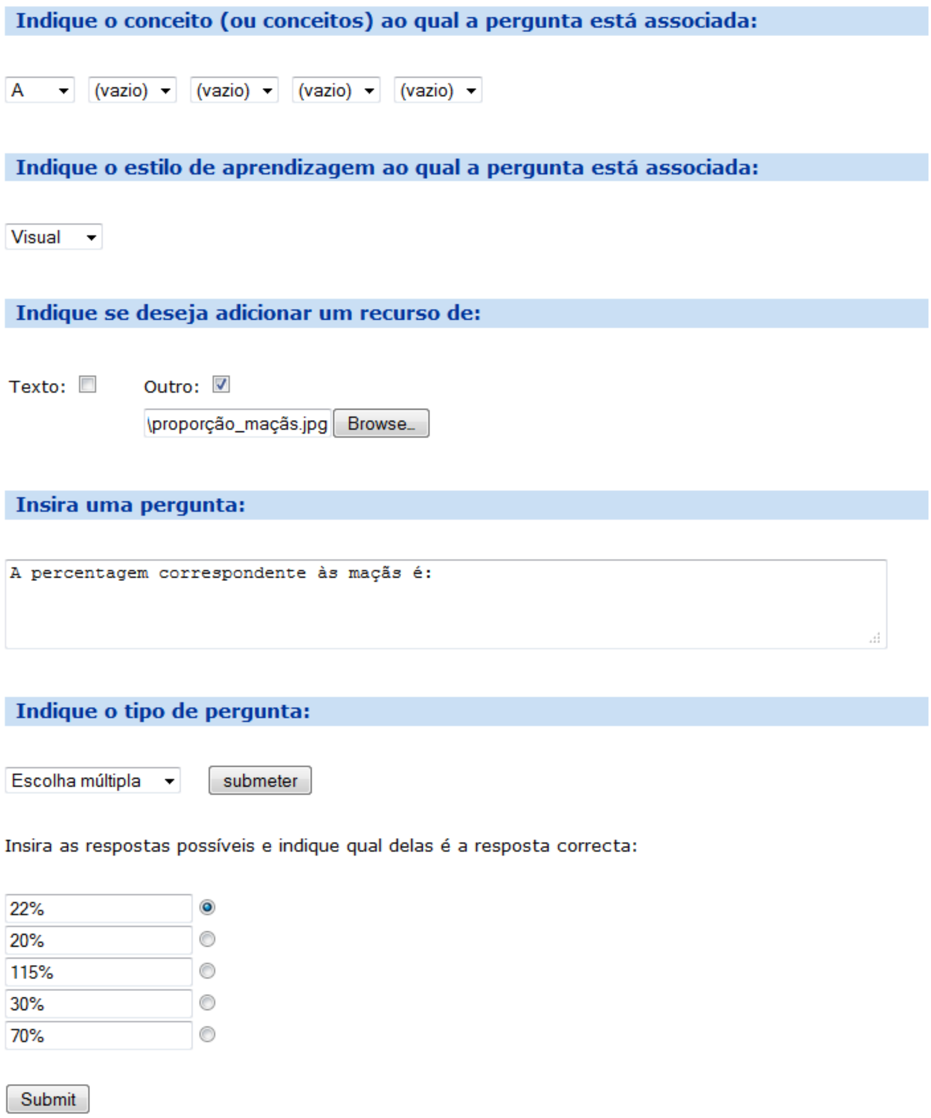

Figure 14 – Interface for the creation of activities.

The user must decide whether the activity will be a multiple choice activity or an open-ended activity. In the case of open-ended activities, the correct answer can be given in several different ways, that is to say the word order may differ. It is therefore necessary to parse the answer, using a probabilistic natural language parser, to verify if it is correct or not.

The activities in the repository may be of two types: simple or parameterized. Activities of the first type are made up of a single body of text with all the parameters in the text already defined. On the other hand, the body of text of parameterized activities doesn't have defined parameters. The author of an activity substitutes all or some of the possible parameters by variables and defines several sets of parameters.

The creation of parameterized activities (Figure 15) requires the user to comply with certain rules. For instance, the user can only utilize a maximum of five different variables, which must have specific names and structure. This makes it possible to find the variables in the text and replace them by the given parameters. A table is provided for the insertion of parameter values, thus ensuring each parameter will be correctly associated with each variable.

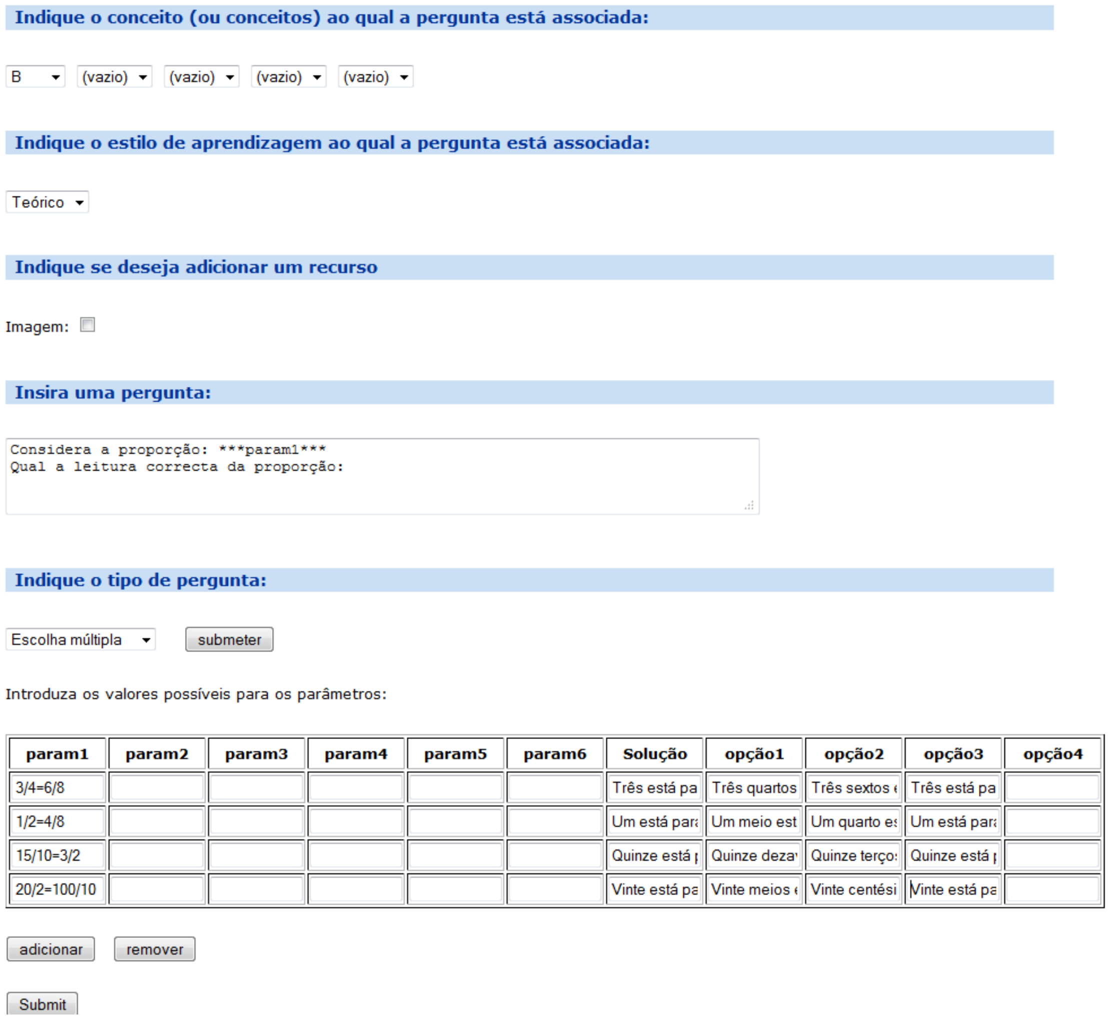

Figure 15 – Interface for the creation of parameterized activities.

The activities shown to each student are kept in a repository of learning objects. Choosing an activity appropriate to the student's current position in the learning path depends on several restrictions. Activities are presented within the context of one or more domain concepts and must therefore be in accordance with those concepts. They must also conform to the student's current dominant learning style. To guarantee such restrictions are met and only appropriate activities are chosen, the adaptation engine obtains that information from AHA's adaptation model. In that way, the activities presented to the student are completely individualized to his personal characteristics and position in the learning path.

If the system chooses an activity which is parameterized, it then retrieves all possible sets of parameters for that activity and randomly chooses one. The system uses the parameters in the set to instantiate each of the variables in the body of text and shows the user an activity with defined parameters. This functionality has the advantage of allowing the creation of various activities/tasks based on a common structure. Instead of manually creating several different activities, the user only has to define its body, which will be common to all activities, and indicate the parameters that will be different in each one. This feature was implemented after observing that Mathematics activities about the same subject are often very similar, with only different parameters.

In addition, the system maintains a record of which activities the student has already performed and uses it to give priority to the ones that haven't been performed yet. This ensures the student won't be shown the same activity twice, for as long as there are activities/tasks in the repository that he hasn't seen yet. In case a student has already performed all the activities in the repository, which obey the concept and learning style restrictions, the system consults the student's record and chooses the activity with the oldest timestamp. Once the activity has been completed the system updates the student's record by adding a new timestamp to that activity.

If an activity is parameterized, a reference to the specific set of parameters used is also added to the student's record. In this way, the same activity might once again be chosen by the system, but it will be instantiated with a different set of parameters, thus presenting the student with an activity different from the one previously shown.

4 Adaptation in PCMAT

The present chapter describes how adaptation is performed in PCMAT. Section 4.1 presents the rules and mechanisms of the platform’s adaptation model, and section 4.2 explains how the choice and presentation of suitable learning objects is accomplished.

4.1 Adaptation Model

The adaptation model defines the adaptation rules and the interaction mechanisms between the user and the system.

PCMAT uses the features contained in the user model to create a specific domain concept graph, adapted from the domain model, and uses it to provide adaptation that will respond to the student's needs. The initial scheme was set by the Mathematics teachers involved in PCMAT’s development, but the path each student takes in the graph is determined by the interaction with the system using progressive assessment, the student's knowledge and the user's characteristics in the user model.

Adaptation occurs through content adaptation, and changes in the structure of links and in the links’ annotation. Content adaptation is achieved by showing or omitting each of the multiple fragments a course page is composed of, as well as displaying alternative or additional information according to each student’s characteristics. These fragments consist of different learning objects such as exercises, figures and narrative text, among others. Changes in the structure of links and the links’ annotation serve the purpose of guiding the student through the course, towards the most relevant information and away from knowledge that is not appropriate yet.

4.1.1 Adaptation Rules

The path a student takes in the concept graph depends on which adaptation rules are activated. These rules are defined by the attributes associated with each domain concept.

Each domain concept is characterized by the following attributes:

- Name (internal)
- Description
- Label – name shown to the user.
- Access – attribute used to verify if a resource is accessible.
- Suitability – attribute used to define the concept's suitability.
- Knowledge – numeric value used to represent the student's level of knowledge.
- Resource – attribute used to define the resources associated with the concept.

Attributes are in turn defined by a name, a description, a default value that can either be a number or a Boolean value, and a set of adaptation rules related to the concept in question.

An adaptation rule consists of a condition and a list of actions:

- The condition of an adaptation rule is a Boolean expression involving attributes of concepts, or attributes of the user model. The activation of the rule is dependent on the evaluation of this condition.
- The list of actions consists of the actions to be performed if the rule is triggered. These actions might update the values of some concept attributes or attributes of the user model.

The following XML fragment illustrates the structure of an adaptation rule (Code 8):

```
<rule>
 <condition><!-- Condition definition -></condition>
 <rule_effect>
        <concept>
               <name><!-- concept's name -></name>
               <attribute><!-- attribute's name -></attribute
               <value><!-- attribute's value -></value>
               <!-- the attribute's value may also be a Boolean expression
->
        </concept>
 </rule_effect>
</rule>
```

Code 8 – Structure of an adaptation rule.

PCMAT's adaptation rules provide the following functionalities:

- Validation of the student's access to content or activities. Each of these is associated with one or more concepts;
- Update of the student's learning preferences and level of knowledge;
- Presentation of content adapted to the student's learning preferences and knowledge;

- Adaptation of the student's path through the domain concept graph according to his learning preferences and level of knowledge;
- Alteration of the student's learning style based on his performance of the activities suggested.

4.1.2 Adaptation Mechanisms

Each page/resource in PCMAT is related to one or more concepts, as well as to the student's dominant learning style. When a student tries to access a resource the following actions occur:

- The access attributes of every concept related to the resource are accessed;
- The adaptation rules associated with the access attribute of each concept are evaluated.

There are two adaptation rules associated with the access attribute that must always be defined:

1. A rule to update the suitability attribute's value. The suitability attribute is used to define whether a resource is appropriate for a student given his current knowledge;
2. A rule that instantiates the value of the resource attribute. The resource attribute takes value 1 if the student's learning style is auditory, value 2 if it's visual or value 3 if it's kinesthetic. The choice of a resource (selection of a Uniform Resource Identifier) that is in accordance with the concept in question and with the student's (dominant) learning style depends on this value.

This process is illustrated in figure 16:

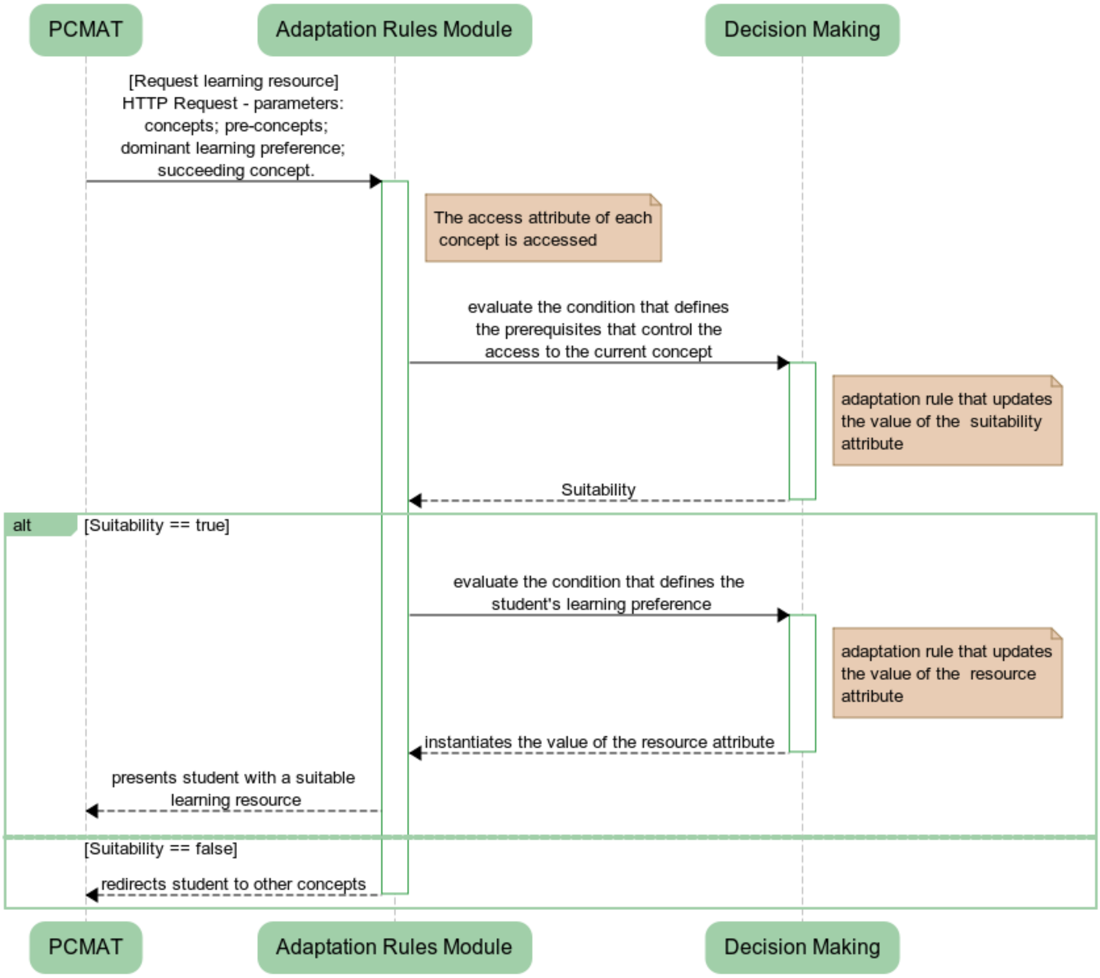

Figure 16 – Evaluation of the adaptation rules associated with the access attribute.

In the case of the first rule, if the suitability attribute takes the value *false* the student can't access the resource and is instead shown another resource, or content, more suitable to his present state. This process is guided by the strategy defined in the GI graph (section 2.3.2), which describes how the student should progress when his learning goals aren't successfully fulfilled.

Should a concept's suitability be t*rue,* the student is able to access the selected resource. In this situation, the concept graph used to steer the student towards new content is the GPCMAT graph (section 2.3.2), which defines how the student should progress in case of success.

The following is an example of an adaptation rule used to update the suitability attribute (Code 9):

```
<rule>
 <condition>
        <!-- conditional expression that defines the pre-requisites
necessary to access the concept in question -->
 </condition>
```

```
  <rule_effect>
        <concept>
              <name>concept_name</name>
              <attribute>suitability</attribute>
              <value>true</value>
        </concept>
  </rule_effect>
</rule>
```

Code 9 – Adaptation rule used to update the suitability attribute.

The next adaptation rule (Code 10) serves the purpose of choosing a resource suitable to the student's dominant learning style. To achieve that, this rule uses the value of the resource attribute, which is defined by one of the adaptation rules of the access attribute.

```
<attribute>
  <name>resource</name>
  <default_value>0</default_value>
  <switch>
        <case>
              <value>1</value>
              <source>URI for auditory content</source>
        </case>
        <case>
              <value>2</value>
              <source>URI for visual content</source>
        </case>
        <case>
              <value>3</value>
              <source>URI for kinesthetic content</source>
        </case>
  </switch>
</attribute>
```

Code 10 – Adaptation rule used to choose a resource suitable to the student's dominant learning style.

The value of the knowledge attribute is an estimate of how much a student knows about a given concept. Updates to this value are based on the student's interaction with the system. Specifically, the value of the knowledge attribute will increase or decrease depending on the student's performance in the activities suggested by the system. The knowledge attribute is used by the adaptation rules responsible for showing and omitting the fragments of content that compose a course's page. It's also used for purposes of link adaptation, namely to guide the user to relevant information and keep him away from unsuitable or irrelevant information. A student will only be able to see content related to a certain concept (or concepts) if he has attained the required minimum knowledge of the concepts that precede it (pre-concepts).

The system has a constructivist approach in the sense that it recommends new content and activities based on the student's previous knowledge and performance in prior activities. To ensure the student won't be presented with the same activity more than once (as long as that's possible), the system maintains a record of which activities the student has already performed and uses it to give priority to activities that haven't been performed yet. In case a student has already performed all of the available exercises, which obey the concept and learning style restrictions, the system consults the student's record and chooses the activity with the oldest timestamp.

When an activity is successfully performed, the knowledge attribute of each of the related domain concepts is updated using the following mechanism (Code 11):

```
Let A1,A2,A3,...,An be the set of concepts associated with the activity
For each i in {1,2,3,...,n}
Ai.knowledge = min(Ai.knowledge+Ai.knowledge*0.25, 100)
```

Code 11 – Mechanism used to update the knowledge attribute.

A similar update is performed for the concepts that, in the concept graph, precede the concepts to which the activity is related (Code 12).

```
Let B1,B2,B3,...,Bm be the set of concepts from which concept Ai depends
For each i in {1,2,3,...,m}
Bi.knowledge = min(Bi.knowledge+Bi.knowledge*0.1, 100)
```

Code 12 – Mechanism used to update the preceding concepts.

The mechanism used to update the knowledge attribute when a student is unable to successfully complete an activity is similar to the one used in case of success, but in this instance the knowledge value is decreased instead.

Updates to the student's learning styles also depend on his performance in the proposed activities. The student's dominant learning style is thought to be correctly inferred whenever an activity corresponding to that learning style is successfully performed. In this situation, that belief is reinforced by incrementing the value of the learning style related to the activity. On the contrary, if the activity isn't successfully completed the value of its learning style, which is also the student's dominant learning style, is decreased and the learning style with the second highest value is increased. The mechanism used to perform this update in case of success is as follows:

- When the learning style related to the suggested activity is personal.lst (auditory) (Code 13):

```
personal.lst = min(personal.lst + 1, 10)
If personal.lsv >= personal.lsp
        then personal.lsv = max(personal.lsv – 1, 0)
If personal.lsv < personal.lsp
        then personal.lsp = max(personal.lsp – 1, 0)
```

Code 13 – Mechanism used to update the auditory learning style.

- When the learning style related to the suggested activity is personal.lsv (visual) (Code 14):

```
personal.lsv = min(personal.lsv + 1, 10)
If personal.lst >= personal.lsp
        then personal.lst = max(personal.lst – 1, 0)
If personal.lst < personal.lsp
```

```
        then personal.lsp = max(personal.lsp – 1, 0)
```

Code 14 - Mechanism used to update the visual learning style.

- When the learning style related to the suggested activity is personal.lsp (kinesthetic) (Code 15):

```
personal.lsp = min(personal.lsp + 1, 10)
If personal.lsv >= personal.lst
        then personal.lsv = max(personal.lsv – 1, 0)
If personal.lsv < personal.lst
        then personal.lst = max(personal.lst – 1, 0)
```

Code 15 - Mechanism used to update the kinesthetic learning style.

The mechanism used to update the student's learning styles when he doesn't successfully complete a task is identical, the difference being that the value of the learning style to which the activity is related is decreased and the learning style with the second largest value is increased.

The changes performed to a student's learning styles, according to his performance, agree with the idea that students may have multiple context dependent learning styles [Fleming, 2007][Miller, 2001]. Taking that into account, the platform was developed to adapt to the student's changes in dominant learning style, thus providing him with appropriate learning objects at all times.

4.1.3 Module for the assessment of open-ended questions

During the development of PCMAT's adaptation engine it became clear assessing open-ended questions would be more challenging than initially expected. The correct answer to these exercises can be given in several different ways, that is to say its syntax may differ. To address that issue a probabilistic natural language parser, the Stanford Parser [Stanford NLP, 2015], is used in conjunction with a Portuguese language grammar [Branco and Silva, 2004][LX-Center, 2009].

Processing a sentence requires it to be previously tagged and tokenized. However, the Stanford Parser isn't capable of tokenizing text written in Portuguese. For this reason, a tokenizer and a part-of-speech tagger have also been developed. The tokenizer separates a sentence into a list of words/tokens, removing punctuation marks in the process. The part-of-speech tagger then assigns to each token a tag indicating which part of speech (category of words with similar grammatical properties) it belongs to.

The Stanford Parser and the Portuguese language grammar allow the system to evaluate answers written in a correct manner. In practice, however, this doesn't always occur. Answers given by children in particular are often written in an incomplete or incorrect form. To address this issue the system uses a set of dependencies between words provided by the Parser, together with a list of object/attribute/value triples. The need for object/attribute/value triples means authors/teachers must provide this information when creating open-ended questions. A student's answer might be incorrectly written, but if the sentence used includes the words contained in the corresponding triple, and their dependency is the same as in the triple, the student's answer is evaluated as being correct. For example, the correct answer to the question "Joana separates used cups of yoghurt from used plastic bottles with a ratio of 9:2. If she separated 45 yoghurt cups, how many plastic bottles did she separate?" is "Joana separated 10 plastic bottles". However, young students will likely answer by simply writing "10 plastic bottles", or "10 bottles", or even just "10". If the system expected students to answer with a correct and complete sentence, it would consider such answers as being incorrect when in fact they are not. However, by using the triple bottles/plastic/10 and the word dependency provided by the Parser, the system is capable of recognizing the above sentences as correct because the dependency between words will agree with the words relationship indicated by the triple. The system is even capable of correctly evaluating the answer when it is just "10", because according to the triple "10" is the expected value. The use of triples enables the system to validate answers it wouldn't be able to validate by using the Parser and the Portuguese grammar alone.

4.2 Learning Objects Recommendation

This section addresses how the choice and presentation of suitable learning objects is performed.

4.2.1 Recommendation Module

The proper choice of learning objects is crucial to PCMAT's adaptability and the individualization of the learning process. In order to choose the most appropriate type of learning object for a given student, in a given section of his learning path, it's first necessary to map the relationship between certain student characteristics and specific parameters of a learning object. To accomplish that task, PCMAT has a recommendation module [Fernandes et al, 2013] that takes as input data from the user model and uses Fuzzy Logic to output a set of parameters the learning object is required to comply with. These parameters are based on elements of the IEEE LOM's general and educational categories [IEEE, 2002].

The input data includes domain dependent data, such as the knowledge the system assumes the student has on the domain, and domain independent data, namely the student's learning style and learning rate (defined as the ratio of number of correct answers to total number of answers). These characteristics are mapped into the following parameters [Friesen et al, 2004]:

- difficulty - indicates the level of ease associated with the use of the learning resource.
- resource type - indicates the potential educational use(s) or type(s) of content associated with the learning resource.
- semantic density - indicates the degree of concision or brevity of expression in a resource.

- interactivity level - indicates the degree to which the learning resource is able to respond to the actions and input of the user.
- interactivity type - indicates whether the resource requires action on the part of the user.

The relationships established between user model characteristics and learning object parameters are the following:

knowledge + learning rate -> difficulty
learning style + learning rate -> resource type
knowledge + learning rate -> semantic density
learning style -> interactivity level
learning style -> interactivity type

In our understanding, both the knowledge level and learning rate should have an influence on the choice of the difficulty level of a learning object. The influence of the student's knowledge level is obvious, but the learning rate should also be taken into account since a student that learns at a faster rate should be able to understand the contents of a learning object with a high degree of difficulty more easily than a student that learns at a slower rate.

The choice of resource type must be constrained by the student's learning style. For example, if a student's learning style is visual then the learning object should be of an appropriate type, such as a diagram or a figure. The learning rate must be considered as well because certain resource types, such as exercises, might at some point in the course be appropriate for faster learning students, whereas slower learning students might need more learning time before being presented with a learning object of that type.

The semantic density of a learning object can be determined in two different ways. It might refer to the ratio between the number of written or spoken words and the total number of words, or it may be determined by the total length of the learning object [Friesen et al, 2004]. The student's knowledge level and learning rate are taken into consideration when determining the appropriate semantic density of a learning object because not only will it be easier for a more knowledgeable student to understand a learning object of greater semantic density, but a student who learns faster is one who understands content more rapidly and therefore should be able to deal with greater semantic density more easily.

As for the interactivity level and interactivity type of a learning object, we have chosen to only factor in the student's learning style because we believe neither knowledge nor learning rate must influence the interactivity of a learning object. A student’s learning style, on the other hand, should be taken into consideration because a highly interactive object seems more appropriate to a student with a kinesthetic learning style, than to a student with an auditory learning style.

The mapping between student characteristics and learning object parameters is performed using Fuzzy Logic. More specifically, the recommendation module uses jFuzzyLogic [Cingolani and Alcala-Fdez, 2012], a library written in Java that uses Fuzzy control language (FCL) to program Fuzzy systems. The recommendation module takes the numeric values, which represent the input data and, after fuzzifying them, uses the specified Fuzzy rules to determine the output parameters with which the learning object must be in accordance. An example of a Fuzzy rule is (Code 16):

```
if learning_rate is slow and knowledge_level is low then difficulty is
very_easy
```

Code 16 – Example of a Fuzzy rule used by the recommendation module.

In a typical Fuzzy program, an output parameter is defuzzified from the linguistic representation of an entire Fuzzy set into a single numeric value. However, learning objects in PCMAT have linguistic parameters, therefore the platform uses the linguistic representation of the Fuzzy system's outputs and not their numeric values. For example, let us consider PCMAT had to present a suitable learning object to a student with the following characteristics: learning style = visual; knowledge = 70%; learning rate = 0.8. The Fuzzy system of the recommendation module would output the parameters: difficulty = difficult; semantic density = high; interactivity type = mixed; interactivity level = medium. Such parameters, as well as a set of context-dependent keywords, are then used by PCMAT's search and retrieval module to retrieve a list of compliant learning objects.

After obtaining the list, the recommendation module verifies in the student model if the object at the top of the list has already been presented to the student. If there is a record of that object in the student model, the system checks the following objects until it cannot find a match. If all the learning objects in the list have already been shown to the student, the recommendation module asks the search and retrieval module for more learning objects that comply with the parameters specified. It then checks the student model again until it finds an object in the list that has not been shown to the student yet. If, after asking the search and retrieval module for learning objects a given number of times, no such object can be found, the system searches in the student model for the learning object with the oldest timestamp. Once the system finds a learning object that can be presented to the student, be it a brand new one or one retrieved from the student model, that object is processed for inclusion in one of the fragments that make up the course's pages (Figure 17).

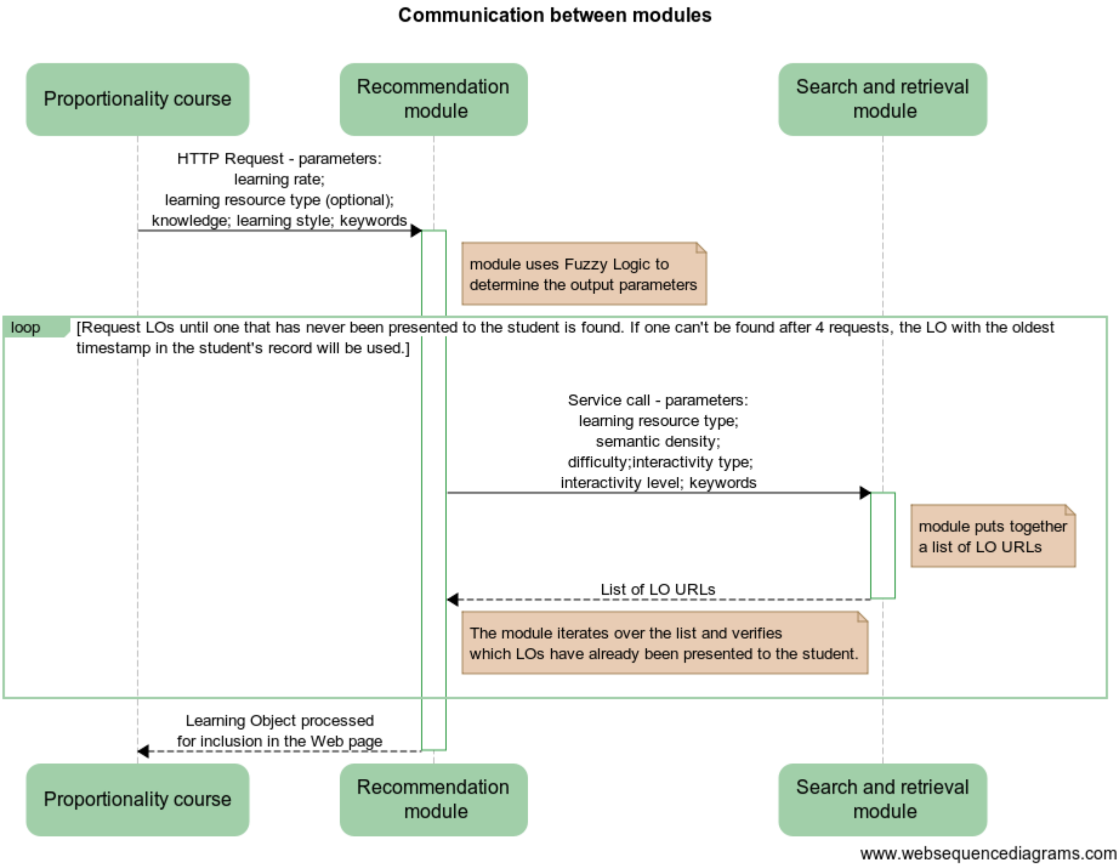

Figure 17 - Communication between modules.

4.2.1.1 Inclusion of learning objects in a Web page

PCMAT's pages are created using XHTML, which means the recommendation module must process a learning object so that it will extend the Web page seamlessly. Learning objects might be of different types, such as images, text, video, and so on. It's also possible that objects of the same type have different formats. For example, a block of text might be kept in a PDF file or in a Word file. Processing distinct types of objects for inclusion in a course's Web pages requires using several techniques. There isn't a single, simple way of including different types of files into an HTML page, especially considering that in many cases browsers don't function in the same manner. For instance, different video formats require different plugins to work. Some plugins, like VLC's Web Plugin, make it possible to play a vast array of video formats in many of the existing browsers, but users are required to install the entire software package and not just the Web plugin. The "video" tag introduced in HTML5 was meant to standardize the way in which video files are incorporated into Web pages. However, at the moment only three video formats are supported and not all browsers are capable of playing all of them.

In view of these issues, in order to embed a variety of learning objects in PCMAT's Web pages different solutions had to be researched and used. To allow video playback, users are told they must install VLC's Web Plugin if it isn't already installed in their system. This plugin was chosen because even though it requires the installation of VLC's entire software package, it enables the playback of several different video formats and works with most browsers.

Files created with Microsoft Office don't open directly in Web browsers. The user is usually asked if he wants to download the file or open it using the program it was created with. However, for the purposes of PCMAT all learning objects must be directly embedded into a course's Web pages, regardless of their type and format. A solution to this problem was found by using Google Docs Viewer, which allows MS Office files to be viewed directly in a Web page without having to use another program. To do this, the learning object's URL must be added to Google Docs Viewer's own URL, which in turn must be used as the source of an HTML "iframe" tag.

Certain file types, such as HTML and TXT, require the use of JQuery in order to dynamically adjust the height of the Web page's fragment in which the learning object is inserted. This adjustment is necessary so that learning objects of these types can be integrated seamlessly into a Web page. JQuery is also used to adjust other parameters related with content format.

4.2.2 Search and Retrieval Module

After the student's domain knowledge, learning style and learning rate have been converted into the IEEE LOM's features the learning object must possess, the recommendation module could search among all the XML documents containing the description of learning objects and select the most appropriate ones. However, the efficiency of this process is subject to the number of documents in the repository and as such, in order to optimize the search for learning objects, the choice was made to use a k-d tree where pointers to the objects' location in the repository are stored. The algorithm used to find the learning objects whose features most resemble the features proposed by the recommendation module is the k-nearest neighbors algorithm [Fernandes et al, 2012].

The k-d tree used by the search and retrieval module [Fernandes et al, 2012] is a 5-dimensional tree, in which each dimension corresponds to one of the values given to the following five elements of the IEEE LOM standard. The learning objects used by PCMAT must have these elements filled in:

- Interactivity type – may take the values *active*, *mixed* or *expositive*, which have been converted to 0, 1 and 2, respectively.
- Learning resource type – this element can take the values exercise, simulation, questionnaire, diagram, figure, graph, slide, table, narrative text, exam, experiment, problem statement, self-assessment, and lecture. These values were grouped according to their interaction type, resulting in the following categories: active (exam, exercise, experiment, problem statement, questionnaire, self-assessment and simulation), textual (lecture, narrative text and table), and visual (diagram, figure, graph and slide). These categories were in turn converted into the

numeric values 0, 1 and 2, respectively. However, it's stated in the IEEE LOM standard that a learning object can have more than one value for this element. Hence, a numeric value was assigned to each possible combination of categories as well: active and textual = 3; active and visual = 4; textual and visual = 5; active and textual and visual = 6.

- Interactivity level – This element can take the values *very low, low, medium, high, very high*, which were converted to 0, 1, 2, 3 and 4, respectively.
- Semantic density - This element can take the values *very low, low, medium, high, very high*, which were converted to 0, 1, 2, 3 and 4, respectively.
- Difficulty - This element can take the values *very easy, easy, medium, difficult, very difficult*, which were converted to 0, 1, 2, 3 and 4, respectively.

Using the numeric values described above, a balanced k-d tree whose nodes refer to all possible value combinations for the five elements (dimensions) was built. A pointer to the location of a learning object in the repository is placed in the k-d tree upon creation of the object's metadata file. The pointer will be placed in the node whose coordinates match the values assigned to the elements described above.

The search for the neighbors nearest to a given point (coordinates defined by the recommendation module) begins at the root of the k-d tree and moves down the tree recursively. At each node, the algorithm chooses to move to the left or the right side of the k-d tree based on whether the point is smaller or larger than that node. Once a leaf node is reached, it is considered the current best guess for the nearest neighbor. The search algorithm then starts unwinding the recursion and as it moves back up the k-d tree, the Euclidean distance is used to determine if a node is closer to the point being considered than the current best guess.

4.2.2.1 K-D Tree Implementation and Maintenance

PCMAT's development involved the creation of a Mathematics course. However, the platform, and the concept behind its creation, may be used to create other courses.

Each course uses a k-d tree as the data structure where the URI of the learning objects are stored. Every day, Microsoft Windows's Task Scheduler tool runs a script that consults a list of existing courses, and creates and loads into memory a k-d tree for each of them. The k-d tree could be kept in memory indefinitely (for as long as the computer/server remained on), but it was decided this process should be performed daily so that new learning objects, introduced in the meantime into the system, could be added to the k-d tree. The script is also set to run whenever the computer/server is turned on, assuring in this way a new k-d tree will be created and loaded into memory if the computer is turned off for some reason.

The k-d tree is loaded into memory using the Java Remote Method Invocation (RMI) API. This mechanism is activated by the previously mentioned script. The RMI allows objects in one

JVM to invoke methods of objects in another JVM. The JVM might be located in the same host or in different hosts. The RMI is in fact an object-oriented remote procedure call mechanism.

There are three programs involved in a remote procedure call:

- The Client is the program that invokes the remote object's method;
- The Server is the program that owns the remote object;
- The RMI Registry is a name service that maps names to remote objects; Servers use the RMI Registry to register remote objects, while Clients use it to lookup remote objects by name; The RMI Register returns a reference to the remote object back to the Client.

Within PCMAT, the recommendation module acts as the client that invokes the remote method, in particular, the method that returns a reference to the k-d tree with the URI of the learning objects. This reference is then used to ask the search and retrieval module for a list of learning objects that are in accordance with specific parameters. The server is the Java program responsible for creating the remote object that represents the k-d tree and registering it in the RMI Registry. It also provides the client with the methods necessary to access the k-d tree.

5 Evaluation Results

The current chapter describes PCMAT's evaluation process. The results obtained were used by FCT to review the PCMAT project (PTDS/CED/108339/2008). The chapter begins with the depiction of the students that participated in the system's evaluation, as well as an explanation of the study's design, in sections 5.1 and 5.2, respectively. The results of the evaluation process are presented in sections 5.3 to 5.5.

5.1 Description of the subjects involved in the experiment

A prototype version of PCMAT has been tested in two basic schools: school EB2,3 Dr. Ferreira de Almeida in Santa Maria da Feira (school 1), and school EB2,3 S. Lourenço in Ermesinde (school 2). The system was used to teach Mathematics, specifically, the subject of Direct Proportionality. This topic is covered by the mathematics program taught in the 6^{th} grade and is addressed once again in the 7^{th} grade.

The study took place over the course of three weeks, and involved three different classes: one class in school 1 and two classes in school 2. The class from school 1 consisted of 25 students. The other two classes had 17 and 19 students. The students were between twelve and fourteen years old.

Over 85% of students were used to personal computers, using them for Web navigation, playing games and social networking. However, none of the students had any previous experience with AHS.

All relevant information was kept in a Microsoft Excel file and subsequently imported to a statistical analysis program, IBM's SPSS [SPSS, 2015].

5.2 Study Design

The prototype's evaluation process was divided in four stages. During the first stage each class was randomly separated into two groups: an experimental group and a control group. The randomization procedure obeyed the following criteria [Martins et al, 2013][Martins, 2012]:

- The number of students in each group is the same or approximate.
- Excellent, good, average and weak students are distributed in equal proportions (if possible) across each group. In addition to the grading information present in the students' academic profile, their knowledge of the topic in question was assessed by a diagnostics test.
- The number of students with the same learning preferences is similar in each group. A questionnaire test was used to determine the students' learning styles [PCMAT, 2013].
- The distribution by gender is similar for each group.

The results of the first stage are the following:

- Creation of two groups in school 1: an experimental group with 12 students and a control group with 13 students. In school 2 each group (experimental and control) has 18 students. In both schools, a teacher was present to assist each group during the learning process.
- Analysis of the data related to the student's learning styles showed that 46.7% of the ones in the experimental group displayed a visual learning preference. In the control group this learning preference is exhibited by 48.4% of the students (Table 4).

Table 4 – Learning preference by group.

Group / Learning Preference	Auditory	Visual	Kinesthetic
Experimental	20.0%	46.7%	33.3%
Control	19.3%	48.4%	32.3%

In this way, the 30 students in the experimental group learned about Direct Proportionality using PCMAT, while the 31 students in the control group had traditional classes using MOODLE (Modular Object-Oriented Dynamic Learning Environment) has a support tool.

The second stage consisted in using questionnaires to obtain the information necessary to instantiate the variables in PCMAT's Student Model. In stage three, students in both groups were required to take a final paper-delivered test. In the final stage, all students answered questionnaires about the adequacy, usability and acceptance of some of PCMAT's functionalities [Martins et al, 2013][Martins, 2012].

5.3 Final Test Results

The final test yielded the following results:

- The mean grade obtained by the students in the experimental group of school 1 was greater than the mean grade attained by students in the control group. The experimental group's mean was 56.9% (σ=18.7) and the control group's mean was 45.7% (σ=18.5). The difference between means is not statistically significant (p=0,164). However, these values seem to indicate that statistically significant results could be obtained with a larger sample size. Considering the sample values have a normal distribution, the two groups were compared using an independent samples t-test with a significance level of 0.05 (5%).
- In school 2, the mean value obtained by the experimental group in the final test was also superior to the mean value obtained by the control group. The experimental group had a mean value of 60.5% (σ=20.8) and the control group had had a mean grade of 43.1% (σ=24.6). The differences observed between samples are statistically significant (p=0.035). The two groups were compared using an independent samples t-test with a significance level of 0.05 (5%).
- Adding the results of the samples from both schools yields the following results: the mean obtained by the students in the experimental group was 59.1% (σ=19.7) and the mean obtained by the students in the control group was 44.2% (σ=21.8). As can be observed, the mean grade attained by the experimental group is positive and greater than the mean attained by the control group. This difference is statistically significant (p=0,010). The two groups were compared using an independent samples t-test with a significance level of 0.05 (5%).

This analysis was performed on the global results of the final test; however, a similar analysis was performed for each domain concept as well. For the purpose of working with a larger sample size, the results of both schools were added together. The comparison between groups was performed using the Mann-Whitney nonparametric test because in this case the samples are non-normal. The samples were found to be non-normal ($p<0.05$) after applying the Kolmogorov-Smirnov test (Table 5).

Table 5 - Results of the Kolmogorov-Smirnov test applied to each concept.

Concept	A1	A2	A3	A4	B1	B2	B3	B4	B5	Final test
p	0.017	0.001	0.000	0.006	0.000	0.000	0.0015	0.001	0.000	0.691

According to the results of the final test, the knowledge acquisition of the students in the experimental group surpassed that of the students in the control group. The results presented in table 6 reveal that this was the case for all of the domain concepts. Statistical analysis, however, shows that only the results concerning concepts A2, A4 and B4 are statistically significant ($p<0.05$) with p-values of p=0.036, p=0.005 and p=0.020, respectively.

Table 6 - Statistical comparison for each concept (Mann-Whitney test).

	Groups	Mean	Ranking	P
A1	Control	43.69	27.54	0.671
	Experimental	50.60	29.33	
A2	Control	57.69	23.85	0.036
	Experimental	79.00	32.53	
A3	Control	28.67	15.60	0.381
	Experimental	43.33	18.17	
A4	Control	15.53	12.23	0.005
	Experimental	57.39	20.97	
B1	Control	72.12	26.23	0.188
	Experimental	86.67	30.47	
B2	Control	31.92	26.85	0.410
	Experimental	42.67	29.93	
B3	Control	60.00	25.58	0.198
	Experimental	72.33	31.03	
B4	Control	35.58	23.35	0.020
	Experimental	63.33	32.97	
B5	Control	8.65	26.69	0.246
	Experimental	19.17	30.07	

The mathematics teachers involved in this project have identified concepts A4, B4 and B5 as the ones students have the most difficulty with. The mean results obtained by the students who used PCMAT were positive for concepts A4 and B4, but negative for concept B5. The students that didn't use the platform didn't perform well on all three concepts, having had negative results.

With the exception of concepts B2 and B5, the mean results attained by students in the experimental group were positive. The students in the control group, however, only obtained positive results on concepts A2, B1 and B3.

The statistically significant results obtained with the final test confirm the hypothesis that the association between PCMAT's several models (Student Model, Domain Model and Adaptation Model), as well as the functionalities it provides, help students obtain better results. The non-statistically significant results also seem to point in that direction. Nonetheless, further experiments using larger sample sizes might help confirm PCMAT's usefulness.

5.4 Learning Preferences

PCMAT has the ability to observe and adapt to the learning preferences of each student. As can be seen in Table 7, no noteworthy changes in learning preferences were observed during the period the students used PCMAT to learn about Direct Proportionality.

Table 7 – Changes in learning preferences.

	Initial learning preference	**Final learning preference**
Auditory	20.0%	16.7%
Visual	46.7%	53.3%
Kinesthetic	33.3%	30.0%

This data came from a questionnaire students were asked to answer so that the information necessary to instantiate the variables of the student model could be obtained. The figures on the right-most column of the table correspond to the data recorded in the student model at the end of the experiment. The dominant learning preference of two students changed from auditory to visual, and a student who initially displayed a kinesthetic learning preference ended the experiment with an auditory learning preference.

These results demonstrate the validity of the process used to assess a student's learning preferences, and of the mechanisms used to monitor the possible changes in dominant learning preference.

The traditional classes attended by the students from the control groups were not adapted to their individual learning preferences.

Although it hasn't been possible to statistically verify that there are benefits to adapting the learning process to each student's learning preferences, the Mathematics teachers involved in this project believe PCMAT's ability to do so is the reason why the students in the experimental groups performed better than the students in the control groups.

5.5 Results of the assessment questionnaires

The assessment questionnaires had the purpose of ascertaining the students' opinion of PCMAT, particularly where its adequacy, usability and acceptance were concerned. In this section, we will begin by presenting the results related to PCMAT's acceptance. We will follow with results about the usefulness and difficulty of use of each of its functionalities, and finish by presenting information about how frequently the students used PCMAT and how it should be used to teach other subjects.

5.5.1 Acceptance and usability

PCMAT's acceptance was evaluated on a scale of 1 to 4 (Table 8). The students filling in the questionnaire could also choose a "no opinion" option.

Table 8 - Scale used to assess PCMAT's acceptance and usability

	Scale
Completely disagree	1
Mostly disagree	2
Mostly agree	3
Completely agree	4
No opinion	0

The results obtained with the assessment questionnaire revealed great acceptance of PCMAT by the students in both schools (Table 9). It was answered by all the students who used the platform (12 in school 1 and 18 in school 2).

PCMAT's usefulness in helping students learn outside of the classroom was determined with the following affirmation: "PCMAT facilitates the study of the subject matter outside of the classroom". Analysis of the options chosen by the students showed that 83.4% of the students from school 1 and 100% of the students form school 2 agree with this affirmation. Only 59.0% of the students from school 1 believe PCMAT helped them study during classes. The percentage of students from school 2 who share this opinion is 93.3%. The lower percentage obtained in school 1 can be explained by the high percentage (41%) of students that chose the "no opinion" option.

Regarding the affirmations "PCMAT is easy to access", "The information provided by PCMAT is appropriately organized, and therefore it's easy to find what one's looking for", "It's easy to access PCMAT's content" and "It's easy to access PCMAT's activities/tasks", analysis of the questionnaire's results showed that more than 65% of the students from both schools chose favorable options (Table 9). This data is a positive indicator of PCMAT's adequacy, usability and acceptance by the students.

The statement "I would recommend the use of PCMAT" also garnered a majority of favorable opinions, with 91.7% of students from school 1 and 93.4% of school 2 stating they would recommend the platform (Table 9). Similarly, 83.4% of students from school 1 and 100% of students from school 2 chose either "completely agree" or "mostly agree" when deciding if they "would like all teachers to use PCMAT in their classes". These results reinforce the assumption that PCMAT was well accepted by the students.

Concerning whether "PCMAT helps improve your performance in the subject matter", 91.7% of participants from school 1 and 93.4% from school 2 answered affirmatively. These results show students have a favorable opinion in regard to the platform's contribution to their performance in Mathematics.

Finally, the results of the assessment questionnaire weren't so positive where PCMAT's graphic design is concerned. Twenty five percent of students from school 1 stated they disagree with the affirmation "PCMAT has a good graphics" (Table 9), a sign that this aspect of the platform needs to be improved.

Table 9 - Results concerning the acceptance and usability of PCMAT.

	Completely disagree		**Mostly disagree**		**Mostly agree**		**Completely agree**		**No opinion**	
	S 1	**S 2**	**S 1**	**S 2**	**S 1**	**S 2**	**S 1**	**S 2**	**S 1**	**S 2**
PCMAT facilitates the study of the subject matter outside of the classroom	0.0	0.0	8.3	0.0	33.4	33.3	50.0	66.7	8.3	0.0
PCMAT facilitates the study of the subject matter during classes	0.0	0.0	0.0	6.7	17.0	46.6	42.0	46.7	41.0	0.0
The information provided by PCMAT is appropriately organized, and therefore it's easy to find what one's looking for	0.0	0.0	16.7	0.0	25.0	20.0	58.3	73.4	0.0	6.6
PCMAT has good graphics	0.0	0.0	25.0	6.6	33.3	53.4	33.4	40.0	8.3	0.0
PCMAT is easy to access	0.0	0.0	0.0	0.0	25.0	40.0	66.7	53.4	8.3	6.6
It's easy to access PCMAT's content	0.0	0.0	0.0	0.0	41.7	40.0	50.0	60.0	8.3	0.0
PCMAT helps improve your performance in the subject matter	0.0	0.0	8.3	0.0	58.4	46.7	33.3	46.7	0.0	6.6
I would recommend	0.0	0.0	8.3	0.0	58.4	20.0	33.3	73.4	0.0	6.6

the use of PCMAT										
I would like all teachers to use PCMAT in their classes	0.0	0.0	8.3	0.0	33.4	33.3	50.0	66.7	8.3	0.0
It's easy to access PCMAT's activities/tasks	0.0	0.0	0.0	0.0	25.0	40.0	58.4	60.0	16.6	0.0

5.5.2 Usefulness and difficulty using PCMAT's functionalities

A scale of 1 to 4 was used to ascertain the usefulness of each of PCMAT's functionalities (Table 10). The user could also choose the option "not familiar with" in case he wasn't familiar with a given functionality.

Table 10 - Scale used to assess the usefulness of PCMAT's functionalities.

	Scale
Useless	1
Not very useful	2
Useful	3
Very useful	4
Not familiar with	0

The scale used to assess the difficulty in using PCMAT's functionalities can be found in table 11. The user may choose the option "not familiar with" in this case as well.

Table 11 - Scale used to assess the difficulty in using PCMAT's functionalities.

	Scale
Difficult	1
Not very easy	2
Easy	3
Very easy	4
Not familiar with	0

These questionnaires were answered by all the students in the experimental groups of both schools (12 in school 1 and 18 in school 2).

In regards to usefulness, the functionality with the best results was "activities/tasks", with 100% of students from both school 1 and school 2 considering it to be either useful or very useful (Table 12).

Table 12 - Results concerning the usefulness of PCMAT's functionalities.

	Useless		**Not very useful**		**Useful**		**Very useful**		**Not familiar with**	
	S 1	**S2**	**S 1**	**S2**	**S 1**	**S2**	**S 1**	**S2**	**S 1**	**S2**
Content index	0.0	0.0	0.0	73.4	75.0	0.0	16.7	26.6	8.3	0.0
Content	0.0	0.0	0.0	0.0	50.0	33.3	50.0	66.7	0.0	0.0
Activities/tasks	0.0	0.0	0.0	0.0	41.7	40.0	58.3	60.0	0.0	0.0
Links	0.0	0.0	8.3	13.3	66.7	53.4	25.0	33.3	0.0	0.0

The functionality "activities/tasks" also obtained the best results in the difficulty assessment; 100% of students from school 1 and 93.4% of students from school 2 considered the functionality easy to use (Table 13).

Table 13 - Results concerning the difficulty of use of PCMAT's functionalities.

	Difficult		**Not very easy**		**Easy**		**Very easy**		**Not familiar with**	
	S 1	**S2**	**S 1**	**S2**	**S 1**	**S2**	**S 1**	**S2**	**S 1**	**S2**
Content index	0.0	0.0	0.0	13.3	66.7	53.4	25.0	33.3	8.3	0.0
Content	0.0	6.7	0.0	6.6	66.7	53.4	33.3	33.3	0.0	0.0
Activities/tasks	0.0	0.0	0.0	6.6	41.7	73.4	58.3	20.0	0.0	0.0
Links	0.0	13.3	0.0	13.4	41.7	46.7	33.3	26.6	25.0	0.0

5.5.3 Frequency of usage

In school 1, 33.3% of students used PCMAT at least once a week outside of the classroom, and 66.7% used it more than once per week. In school 2, the platform was used at least once a week by 53.3% of the students, and more than once a week by 40% of them (Table 14).

Table 14 – Frequency of usage of PCMAT.

Frequency of usage	School 1	School 2
Never	0.0%	0.0%
At least once a week	33.3%	53.3%
More than once a week	66.7%	40%
Every day, or almost every day	0.0%	0.0%

Concerning the question "Would you like other subjects to function with PCMAT?", 91.7% of students from school 1 and 100% of students from school 2 answered affirmatively. These results are in concordance with the opinions displayed in regards to the statement "I would like all teachers to use PCMAT in their classes". This affirmation gathered 83.4% of favorable responses from school 1 and 100% from school 2.

6 Conclusion and Future Work

In this chapter, the research and development performed under the scope of this thesis is reviewed and discussed. In section 6.1 the work described in the present document is concluded, and in section 6.2 its limitations and possible solutions are examined.

6.1 Conclusion

The PCMAT project had the purpose of developing and testing an Adaptive Educational Hypermedia System, aimed at assisting students with the study of Mathematics in the context of basic schools. This was accomplished by combining Adaptive Hypermedia techniques with constructivism and learning styles to provide innovative content adaptation. In addition, the work performed resulted in the definition of a new architecture and strategies for the implementation of an AEHS to support and improve Mathematics in basic schools. The PCMAT platform was also developed in an attempt to contribute to the progress of AEHSs, in particular where adaptation techniques are concerned. As e-learning systems become more commonplace and grow in prominence, the usefulness of adaptive systems becomes more apparent.

PCMAT's adaptation model establishes a set of adaptive and dynamic pedagogical strategies, as well as the interaction mechanisms between the user and the system. It not only provides adaptation, such as content adaptation and link adaptation, but does so by taking into account the constructivist learning theory and the learning styles theory. The system continuously adapts to the student's learning style in an attempt to achieve the best possible results. As the student's learning style changes, so does the content proposed by the system. The system has a constructivist approach in the sense that it recommends new content and activities based on the student's previously acquired knowledge and performance in prior activities, guiding him towards appropriate content and helping him integrate and assimilate newly acquired knowledge.

Among PCMAT's accomplishments is the definition of the processes and tools necessary to produce learning objects aligned with an adopted standard. PCMAT uses learning objects that comply with the IEEE LOM metadata standard. In this way, learning objects created for the PCMAT platform can be shared across learning object repositories, and be used by other learning platforms.

In order to present students with appropriate learning objects the system uses Fuzzy Logic to define the relationship between specific student characteristics and the parameters of a learning object. These parameters might not exactly match any of the learning objects in the repository. However, by using a k-d tree and the k-nearest neighbors algorithm the system is capable of finding the learning object that most resembles the "ideal" learning object. Since it isn't viable to create specific learning objects for each student's particular combination of characteristics, these mechanisms allow the choice of adequate learning objects under conditions of uncertainty.

We have also succeeded in defining, implementing and validating a Student Model that not only describes the personal information, knowledge, preferences, and learning styles of the user, but also supports adaptive functionalities based on the use of the IEEE LOM standard.

PCMAT was tested in two basic schools with positive outcomes. The results show the average student scores, from both schools, in the experimental group was higher than the average student scores in the control group, 59,1% (σ = 19,7) against 44,2% (σ = 21,8). The differences observed are statistically significant (p=0,010). Students from the experimental groups also performed better in the knowledge acquisition of individual concepts.

These results are very positive, and a strong indicator that PCMAT's architecture is viable and appropriate for AEHSs used in the context of basic schools. They also allow us to conclude that AEHSs, by adapting to the different needs and characteristics of students, contribute indeed to the effectiveness and efficiency of the learning process. In addition, students perceived this tool as being relevant to their learning experience, and are of the opinion that the platform should be integrated in a more global learning strategy that also includes tutoring (direct contact with the teacher) and peer-learning. The teachers that participated in this experiment agreed with these definitions of the platform as well.

6.2 Limitations and Future Work

The results obtained with PCMAT allow us to conclude our goals have been successfully accomplished. Nonetheless, there are several aspects of the platform that could be improved.

As indicated by the students' answers to the assessment questionnaires, PCMAT's graphical interface needs to be improved. The platform's functionalities were given priority and consequently the user interface was somewhat neglected. This is, however, an important part of the system that can positively contribute to the student’s learning experience and should therefore be updated.

The results obtained during the testing phase could also be confirmed and reinforced if the system was tested anew using a larger sample size. While several aspects of the platform garnered statistically significant results, the results obtained for some features were inconclusive. The use of learning styles is one such case, since we weren’t able to conclusively demonstrate that the positive results obtained with the platform were a direct result of their application. It can be concluded that adapting content and activities to each student’s characteristics and knowledge yields better results than a more traditional learning method, but that improvement cannot be specifically attributed to the use of learning styles. More work and experimentation is necessary before that affirmation can be made.

Other areas that need to be worked on include the assessment of open-ended questions using natural language processing, and the collaborative quality of the platform. Regarding the former, the process has been researched and tested but its effective implementation hasn’t been concluded yet. As for the latter, although the word “collaborative” is part of the platform’s name, the system doesn’t yet possess this quality. This limitation will be addressed in future iterations by including a chat environment that will allow students to exchange messages during the learning process. This feature will give students the possibility of sharing their ideas and helping each other. It will also be possible for the system to analyze the messages exchanged between students with the purpose of identifying the difficulties and doubts they might be having, in order to give them appropriate assistance and feedback.

References

[Atkinson and Shiffrin, 1968] Atkinson R. C. and Shiffrin, R. M. (1968) Human memory: A proposed system and its control processes. In *The Psychology of Learning and Motivation: Advances in Research and Theory,* K. W. Spence and J. T. Spence, Eds. Vol. 2, p. 89-195. New York: Academic Press.

[Bandura et al, 1961] Bandura A., Ross D. and Ross S. A. (1961) Transmission of aggression through the imitation of aggressive models. *Journal of Abnormal and Social Psychology*, Vol. 63 (3), p .575–582.

[Bandura, 1977] Bandura A. (1977) *Social learning theory*. Englewood Cliffs, NJ: Prentice-Hall.

[Bauersfeld, 1995] Bauersfeld H. (1995) The Structuring of the Structures: Development and Function of Mathematizing as a Social Practice. *Constructivism in Education*, p 137-158. Lawrence Erlbaum Associates Publishers, Hillsdale, NJ.

[Benyon and Murray, 1993] Benyon D. and Murray D. (1993) Applying user modelling to human-computer interaction design. *Artificial Intelligence Review*, Vol. 7(3), p. 199-225.

[Benyon, 1993] Benyon D. (1993) Adaptive systems: A solution to Usability Problems*. Journal of User Modeling and User Adapted Interaction*, Kluwer, Vol. 3(1) p. 1-22.

[Branco and Silva, 2004] Branco A. and Silva J. (2004) Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004),* ELRA, Paris, p.507-510.

[Brown et al, 1989] Brown J. S., Collins A. and Duguid P. (1989) Situated Cognition and the Culture of Learning. *Educational Researcher*, Vol. 18(1), p. 32-42.

[Brown et al, 2007] Brown E., Fisher T. and Brailsford, T. (2007) Real Users, Real Results: Examining the Limitations of Learning Styles within AEH. *Proc. 18th Conf. Hypertext and Hypermedia*, p. 57-66.

[Brownstein, 2001] Brownstein B. (2001) Collaboration: The foundation of learning in the future. *Education*, Vol. 122(2), p. 240–247.

[Brusilovsky et al, 2007] Brusilovsky P., Kobsa A. and Nejdl W. (Eds) (2007) *The Adaptive Web, Methods and Strategies of Web Personalization*. Springer-Verlag Berlin Heidelberg.

[Brusilovsky, 1996a] Brusilovsky P. (1996) Adaptive hypermedia: an attempt to analyze and generalize. In *Multimedia, Hypermedia, and Virtual Reality. Lecture Notes in Computer Science*, Brusilovsky P., Kommers P. and Streitz N., Eds. Vol. 1077, p. 288–304.

[Brusilovsky, 1996b] Brusilovsky P. (1996) Methods and Techniques of Adaptive Hypermedia. *User Modeling and User Adapted Interaction*, Vol. 6(2), p. 87-129.

[Brusilovsky, 2001] Brusilovsky P. (2001) Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, Vol. 11 (1/2), p. 87-110.

[Chapman, 2003] © Chapman A. (2003) *Howard Gardner's Multiple Intelligences*. [Online] Available from: http://www.businessballs.com/howardgardnermultipleintelligences.htm [Accessed 18th June 2015], Not to be sold or published. Alan Chapman/ Businessballs accepts no liability for

any issues arising.
[Chapman, 2005] © Chapman A. (2005) *VAK learning styles*. [Online] Available from: http://www.businessballs.com/vaklearningstylestest.htm [Accessed 18th June 2015], Not to be sold or published. Alan Chapman/ Businessballs accepts no liability for any issues arising.
[Chapman, 2009] © Chapman A. (2009). *Kolb Learning Styles*. [Online] Available from: http://www.businessballs.com/kolblearningstyles.htm [Accessed 16th June 2015], Not to be sold or published. Alan Chapman/ Businessballs accepts no liability for any issues arising.
[Chepegin et al, 2004] Chepegin V., Aroyo L., De Bra P. and Heckman D. (2004) User Modeling for Modular Adaptive Hypermedia. *Workshop on Semantic Web for E-Learning*, p. 366-371.
[Cingolani and Alcala-Fdez, 2012] Cingolani P. and Alcala-Fdez J. (2012) jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation. *IEEE International Conference on Fuzzy Systems*, p. 1-8.
[Coursera, 2015] Coursera (2015) *About*. [Online] Available from: https://www.coursera.org/about/ [Accessed: 10th September 2015].
[Couto et al, 2011] Couto P., Martins C., Faria L., Fernandes M. and Carrapatoso E. (2011) PCMAT Metadata Authoring Tool. *Proc. ISCIES 2011, International Symposium on Computational Intelligence for Engineering Systems*, p. 355-363.
[De Bra et al, 2003] De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Santic T., Smits D. and Stash, N. (2003) AHA! The Adaptive Hypermedia Architecture. *Proceedings of the ACM Hypertext Conference*, Nottingham, UK, p. 81-84.
[De Bra et al, 2004] De Bra P., Aroyo L. and Chepegin V. (2004) The Next Big Thing: Adaptive Web-Based Systems. *Journal of Digital Information*, Vol. 5(1), article no. 247.
[De Bra et al, 2006] De Bra P., Smits D. and Stash N. (2006) *The Design of AHA!*. [Online] Available from: http://aha.win.tue.nl/ahadesign/ [Accessed 5th April 2014]
[De Bra, 2006] De Bra P. (2006) Web-based educational hypermedia. In *Data Mining in E-Learning*, Romero C. and Ventura S., Eds. Universidad de Cordoba, Spain, WIT Press, p. 3-17.
[dos Santos and Travassos, 2011] dos Santos P. S. M. and Travassos G. H. (2011) Action Research Can Swing the Balance in Experimental Software Engineering. *Advances in Computers*, Vol. 83, p. 205-276.
[Duffy and Jonassen, 1992] Duffy T.M., Jonassen D. (Eds.) (1992) *Constructivism and the technology of instruction: A conversation*. Hillsdale NJ: Lawrence Erlbaum Associates.
[Easterbrook et al, 2008] Easterbrook S., Singer J., Storey M. and Damian D. (2008) Selecting Empirical Methods for Software Engineering Research. *Guide to Advanced Empirical Software Eng.*, p. 285-311.
[edSurge, 2015] edSurge (2015) *Coursera and edX Now Claim Over 24 Million Students*.
[edX, 2015] edX (2015) *About us*. [Online] Available from: https://www.edx.org/about-us [Accessed: 10th September 2015].
[Fernandes et al, 2012] Fernandes M., Couto P., Martins C., Faria L., Bastos C. and Costa F. (2012) Learning objects recommendation in an adaptive educational hypermedia system. *Proc. 8th WSEAS International Conference on Educational Technologies (EDUTE '12)*, p. 123-

128.

[Fernandes et al, 2013] Fernandes M., Couto P., Martins C. and Faria L. (2013) Content Adaptation for an Adaptive Hypermedia System. *Proc. INTELLI 2013, The Second International Conference on Intelligent Systems and Applications*. p. 54-57.

[Ferster and Skinner, 1957] Ferster C. B., Skinner B. F. (1957) *Schedules of Reinforcement*. New York: Appleton-Century-Crofts.

[Fleming and Mills, 1992] Fleming N.D., Mills C. (1992) Not Another Inventory, Rather a Catalyst for Reflection. *To Improve the Academy*, 11, p. 137-155.

[Fleming, 2007] Fleming N. D. (2007) VARK 'A review of those who are multimodal'. [Online] Available from: http://business.vark-learn.com/the-vark-categories/multimodal/multimodal-article/ [Accessed: 1st October 2011]

[Fraley, 2001] Fraley L.F. (2001) Strategic interdisciplinary relations between a natural science community and a psychology community. *The Behavior Analyst Today*, Vol. 2(4), p 290–324.

[Friesen et al, 2004] Friesen N., Fisher S. and Roberts A. (2004) *CanCore Guidelines Version 2.0: Educational Category*. [Online] Available from: http://cancore.tru.ca/guidelines/CanCore_Guidelines_Educational_2.0.pdf [Accessed February 2013]

[Gamoran et al, 2000] Gamoran A., Secada W.G. and Marrett, C.A. (2000) The organizational context of teaching and learning: changing theoretical perspectives. *Handbook of the Sociology of Education*, p. 37-63.

[Good and Brophy, 1990] Good T. L. and Brophy J. E. (1990) *Educational psychology: A realistic approach*. 2nd Ed. Holt McDougal.

[Graham, 2000] Graham G. (2000) *Behaviorism*. [Online] Available from: http://plato.stanford.edu/entries/behaviorism/ [Accessed: 13th June 2015]

[Haggbloom et al, 2002] Haggbloom S. J., Warnick R., Warnick J. E., Jones V. K., Yarbrough G. L., Russell T. M. et al. (2002) The 100 most eminent psychologists of the 20th century. *Review of General Psychology*, Vol 6(2), p. 139–152.

[Hargreaves et al, 2005] Hargreaves D. et al (2005) *About learning: Report of the Learning Working Group*. Great Britains: Demos.

[Hill and Carver Jr., 2000] Hill J.M.D. and Carver Jr., C.A. (2000) *Adaptive Hypermedia Systems*. [Online] Available from: http://www.west-point.org/users/usma1982/39377/john/Publications/2000/2000-10_WebAccessibility/HillCarverAHS.pdf [Accessed 11th March 2014]

[Holt and Willard-Holt, 2000] Holt D. G., Willard-Holt C. (2000) Let's get real – students solving authentic corporate problems. *Phi Delta Kappan*, 82(3), p. 243-246.

[IEEE, 2002] IEEE 1484.12.1-2002. (2002) Standard for learning object metadata. *The Institute of Electrical and Electronics Engineers*, Inc. New York.

[IEEE, 2005] IEEE 1484.12.3.2005 (2005) Standard for learning technology: extensible markup language (XML) schema definition language binding for learning object metadata. *The Institute of Electrical and Electronics Engineers, Inc*. New York.

[Jonassen, 1994] Jonassen D. (1994) Thinking technology. *Educational Technology*, Vol. 34(4), p. 34-37.

[Knutov et al, 2009] Knutov E., De Bra P. and Pechenizkiy M. (2009) AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, Vol. 15(1), p. 5-38.

[Kobsa, 1993] Kobsa A. (1993) User modeling: Recent work, prospects and hazards. In *Adaptive User Interfaces: Principles and Practice*, Schneider-Hufschmidt M., Kühme T., Malinowski U., Eds. Amsterdam: North Holland Elsevier.

[Kolb and Kolb, 2005] Kolb A. and Kolb D. (2005) Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of Management Learning and Education*, 4(2), p. 193-212.

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

[LX-Center, 2009] LX-Center (2009) *Language Resources and Technology for Portuguese*. [Online] Available from: http://lxcenter.di.fc.ul.pt/. [Accessed 30th August 2015]

[Martins et al, 2008a] Martins C., Faria L., Carvalho C. V. and Carrapatoso E. (2008) User Modeling in Adaptive Hypermedia Educational Systems. *Educational Technology & Society*, Vol. 11(1), p. 194–207.

[Martins et al, 2008b] Martins C., Faria L. and Carrapatoso E. (2008) Constructivist approach for an educational adaptive hypermedia tool. *Proc. ICALT*, IEEE, p. 303–305.

[Martins et al, 2011] Martins C., Couto P., Fernandes M., Bastos C., Lobo C., Faria L. and Carrapatoso E. (2011) PCMAT - Mathematics Collaborative Learning Platform, *Highlights in Practical Applications of Agents and Multiagent Systems*, AISC 89, Springer-Verlag, Berlin Heidelberg, p. 93-100.

[Martins et al, 2013] Martins C., Faria L., Fernandes M., Couto P., Bastos C. and Carrapatoso E. (2013) PCMAT Mathematics Collaborative Educational System. In *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends*, Peña-Ayala A., Ed. Vol. 17, Springer, Berlin - Heidelberg, p. 183-212.

[Martins, 2012] Martins C. (2012) *Sistemas hipermédia adaptativa para suporte de ambientes de aprendizagem construtivistas.* [Online] Available from: http://hdl.handle.net/10216/72590 [Accessed 30th August 2015]

[Masters, 2011] Masters K. (2011) A Brief Guide To Understanding MOOCs*. The Internet Journal of Medical Education*, vol. 1(2).

[Miller, 2001] Miller P. (2001) Learning styles: the multimedia of the mind. *Educational Resources Information Center* (ED451140).

[Montgomery and Groat, 1998] Montgomery S., Groat L. N. (1998) *Student Learning Styles and Their Implications for Teaching*. [Online] Available from: http://www.crlt.umich.edu/publinks/CRLT_no10.pdf [Accessed: 26th September 2011]

[Nelson, 1965] Nelson T. (1965) Complex information processing: a file structure for the complex, the changing and the indeterminate. *ACM '65, Proceedings of the 20th national conference*. ACM New York, NY, USA, p. 84-100.

[Neven and Duval, 2002] Neven F. and Duval E. (2002) Reusable learning objects: a survey of LOM-based repositories. *Proceedings of the 10th ACM international conference on Multimedia (MULTIMEDIA '02)*. ACM, New York, USA, p. 291-294.

[NISO, 2004] NISO (2004) *Understanding metadata.* Bethesda: NISO Press.

[OECD, 2014] OECD (2014) *PISA 2012 Results in Focus: What 15-year-olds know and what they can do with what they know*.

[PCMAT, 2013] PCMAT (2013) *PCMAT Registration*. [Online] Available from:

http://pcmat.dei.isep.ipp.pt/pcmatpub/registo.html [Accessed 7th August 2015]

[Prawat and Floden, 1994] Prawat R. S. and Floden, R. E. (1994) Philosophical perspectives on constructivist views of learning. *Educational Psychologist*, Vol. 29(1), p. 37-48.

[Richmond and Cummings, 2005] Richmond A. S. and Cummings R. (2005) Implementing Kolb's learning styles into online distance education*. International Journal of Technology in Teaching and Learning*, Vol. 1(1), p. 45-54.

[Ritu and Sugata, 1999] Ritu D. and Sugata M. (1999) Learning Styles and Perceptions of Self. *International Education Journal*, Vol. 1(1), p. 61-71.

[Skinner, 1953] Skinner B. F. (1953) *Science and human behavior*. Simon and Schuster.

[Smith, 2001] Smith M. K. (2001). *'David A. Kolb on experiential learning', the encyclopedia of informal education*. [Online] Available from: http://infed.org/mobi/david-a-kolb-on-experiential-learning/ [Accessed 1st October 2011]

[SPSS, 2015] SPSS (2015) *SPSS Software*. [Online] Available from: http://www-01.ibm.com/software/analytics/spss/ [Accessed 7th August 2015]

[Stahl, 2002] Stahl S. A. (2002) Different strokes for different folks?. In *Taking sides: Clashing on controversial issues in educational psychology*, Abbeduto L., Ed. Guilford, CT, USA: McGraw-Hill, p. 98-107.

[Stanford NLP, 2015] Stanford NLP Group (2015) *The Stanford Natural Language Processing Group*. [Online] Available from: http://nlp.stanford.edu/software/lex-parser.shtml [Accessed 30th August 2015]

[Susman and Evered, 1978] Susman G. L. and Evered R. D. (1978) An assessment of the scientific merits of Action Research. *Administrative Sciences Quarterly*, Vol. 23, p. 582–603.

[Todes, 2002] Todes D. P. (2002) Pavlov's Physiology Factory. *Baltimore MD: Johns Hopkins University Press*. p. 232-340.

[Tolman and Honzik, 1930] Tolman E. C. and Honzik C. H. (1930). Introduction and removal of reward, and maze performance in rats. *University of California Publications in Psychology,* Vol. 4, p. 257-275.

[Tolman, 1948] Tolman E. C. (1948) Cognitive maps in rats and men. *Psychological Review*, Vol. 55(4), p. 189-208.

[Triantafillou et al, 2003] Triantafillou E., Pomportsis A. and Demetriadis S. (2003) The design and the formative evaluation of an adaptive educational system based on cognitive styles. *Computers & Education*, Vol. 41, p.87-103.

[Udacity, 2011] Udacity (2011) *About Us*. [Online] Available from: http://www.udacity.com/us [Accessed: February 2013].

[VARK, 2015] VARK (2015)*The VARK Modalities*. [Online] Available from: : http://vark-learn.com/introduction-to-vark/the-vark-modalities/ [Accessed 16th June 2015]

[von Glaserfeld, 1989] von Glasersfeld E. (1989) *Constructivism in Education*. [Online] Available from: http://www.vonglasersfeld.com/114 [Accessed: 14th June 2015]

[Watson and Rayner, 1920] Watson J. B. and Rayner R. (1920) Conditioned emotional reactions. *Journal of Experimental Psychology*, Vol. 3(1), p 1-14.

[Watson, 1913] Watson J. B. (1913) Psychology as the Behaviorist Views it. *Psychological Review*, Vol. 20, p. 158-177.

[Wertsch, 1985] Wertsch J.V (1985) *Vygotsky and the social formation of mind*.

Cambridge, MA: Harvard University Press.

[Wiley, 2002] Wiley D. A. (2002) *Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy*. [Online] Available from: http://www.reusability.org/read/ [Accessed 12th August 2015]

[Woodward and Chambers, 1983] Woodward C. A. and Chambers L. (1983) *Guide to Questionnaire Construction and Question Writing*. Canadian Public Health Association.

[Wu et al, 1998] Wu H., Houben G.J. and De Bra P. (1998) AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. *Proc. of the "Zesde Interdisciplinaire Conferentie Informatiewetenschap"*, p. 77-88.

[Wu et al, 2000] Wu H., De Bra P., Aerts A. and Houben G. (2000) Adaptation Control in Adaptive Hypermedia Systems. In *Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science*, Brusilovsky P., Stock O. and Strapparava C., Eds. Vol. 1892, p. 250-259.