Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

Natalia Stash

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Stash, Natalia

Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System / door Natalia Stash.
Eindhoven: Technische Universiteit Eindhoven, 2007. Proefschrift.

ISBN 978-90-386-1050-4
NUR 983

Subject headings: hypertext / hypermedia / adaptive hypermedia / cognitive psychology
CR Subject Classification (1998): H.5.4, H.5.2, H.5.1, H.3.4, H.2.4, J.4

SIKS Dissertation Series No. 2007-11
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by University Press Facilities, Eindhoven, the Netherlands.
Cover design: Paul Verspaget & Carin Bruinink, Nuenen, The Netherlands.
The images were taken from
http://www.jcu.edu.au/studying/services/studyskills/learningst/index.html
with permission from James Cook University (JCU), Australia.

Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus, prof.dr.ir. C.J. van Duijn,
voor een commissie aangewezen door het College voor Promoties
in het openbaar te verdedigen
op maandag 2 juli 2007 om 16.00 uur

door

Natalia Stash

geboren te Sint Petersburg, Rusland

Dit proefschrift is goedgekeurd door de promotoren:

prof. dr. P.M.E. De Bra
en
prof. dr. L. Hardman

Copromotor:
dr. A.I. Cristea

Acknowledgments

Dear reader, the time has come to tell about the research I was doing for my PhD thesis. But before that I would like to express my gratitude to a number of people for different reasons on my long way leading to a PhD.

I would like to thank my first promotor prof. dr. Paul De Bra for being such a special supervisor, for providing an excellent environment in which I was able to develop my skills as a researcher and as a programmer, for being a great source of encouragement in all my research attempts and trials, for giving the most deep and clear explanations to all my questions and for his patience waiting until I finish writing my PhD thesis after I started working on a new project. I would like to express my warm thanks to his wife Rina. The two of them really became my second parents during my stay in the Netherlands!

I would like to thank my second promotor prof. dr. Lynda Hardman for providing extremely valuable comments on my thesis and giving much more inspiration for describing the work done. Implementing her comments in the thesis made it look a lot better!

I would like to thank my co-promotor dr. ir. Alexandra Cristea for being a wonderful supervisor with whom I had a great pleasure to work and with whom I had enjoyable time during our joint trips to various conferences.

I would like to express my gratitude to all the other committee members. Specifically, I would like to thank prof. dr. Tatiana Gavrilova under whose supervision I had a great pleasure to work while I was still in Saint-Petersburg, Russia. It was in her Intelligent Systems laboratory that I became interested in adaptive hypermedia research. Tatiana involved me in various interesting projects and we wrote a number of papers most of which were accepted at different international conferences. I would like to thank prof. dr. Peter Brusilovsky who I met at the First International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH2000) in Trento, Italy in 2000 where I was giving my first presentation at the doctoral consortium. Thank you, Peter, for introducing me to Paul De Bra, to whose group I moved one year after the conference. Moving from Russia to the Netherlands was a big step in my life. Thank you also for our discussions of my work and providing valuable feedback.

I would very much like to thank the supervisor of my diploma at the Saint-Petersburg Electro-Technical University — Elena Kotova. It happened during her lectures that I became interested in the subject of Intelligent Systems and therefore I chose to write a diploma under her supervision. She introduced me to Tatiana Gavrilova in whose laboratory I was working on the diploma and with whom I continued working after the defense.

My work on the PhD thesis was made possible through a grant from the NLnet Foundation and the ADAPT Minerva Project. The first two years I was working in collaboration with a number of students from the Eindhoven University of Technology. Therefore this thesis is largely supported by the results of our joint work. I would specifically like to thank David Smits who developed the core engine of the AHA! system. Without his work many of my extensions to the AHA! system would not have been possible.

I would like to thank a number of learning styles experts who helped me to validate my learning styles extension to the system — Richard Felder, Jan Vermunt, Martin Graff and Daniel Burgos. Thank you for your time and willingness to participate in my experiment and for providing your valuable comments.

I would like to thank all my colleagues from the Information Systems group, specifically our secretary Riet van Buul for her enormous help with various issues. Thank you, Riet, for reading my whole thesis in order to find mistakes! My thanks go to Olivia Oanea who was once my officemate for her help with different technical issues while I was writing my thesis, to Reinier Post for being able to help me with solving the server problems at any time, to Lusine Hakobyan, Katya Vasilyeva (who I knew since Intelligent Systems laboratory in Saint-Peterburg) and her husband Mykola Pechenizkiy for being wonderful friends. Katya, Mykola, I am very happy that you moved to Eindhoven!

I would very much like to thank all my colleagues from the CHIP project. I would like to express my deepest gratitude to Lora Aroyo, the supervisor of the project, for her understanding of my situation of being involved in the project and still having to write the PhD thesis and providing me with all the possibilities to do that. Thank you for all nice activities we had together besides work! I thank Lloyd Rutledge for being a lovely officemate and a constant source of humor. I thank Yiwen Wang for bringing a very special atmosphere to the project. My warm thanks to all the people at the Rijksmuseum involved in CHIP, especially to Peter Gorgels and Xenia Henny.

I would like to thank Olga Tveretina who was the first person to become my friend in Eindhoven for all the nice time we had together in the Netherlands and in Ireland and for all her wise advice! My warm thanks go to Rasa Jurgelenaite, whom I met at the SIKS courses, for being a wonderful friend ready to come to help at any difficult moment of my life. I thank more of my friends who I met at the university: Vadim Chepegin for being able to provide help with any of my research questions; Yulia Arzhaeva, Samuil Angelov and Apostolos Doris — for all the enjoyable time we had together.

I express my deepest gratitude to Guus Angeneind for introducing me to the Dutch culture, for teaching me to speak the Dutch language, for help with all the questions related to living in the Netherlands and support in the difficult months of finalizing my thesis. I thank you, your family and friends for all the warmth I have been getting these years.

Finally, my greatest thanks go to my parents for their constant support throughout my entire life, for finding the most convincing words and their belief in me, for their constant encouragement never to stop after achieving something and to always go further, for the wonderful time we had during our traveling. Thank you for being very special parents!

Natalia Stash
Amsterdam, May 2007

To my parents, Nina and Victor

Contents

Chapter 1

Introduction

Have you ever wondered what it would be like to live in a world where everything is or feels just right, where everything is arranged according to your interests, tastes and needs, where you can do things that you like or need to do without wasting your time on non-interesting and non-relevant things?

You come home and the TV starts showing your favorite programs, you sit in the car and the radio plays music you (and your company) like; you go shopping for clothes and immediately find things you need — as the stores seem to offer everything only in your size and in the style that you like; you wish to get some information and the information services provide you with all the data you need, without distracting you with information you do not want to receive. And, of course, if your tastes and interests change the environment adapts to suit them.

Certainly not all of these things are possible in the real world (at this time). But in the area of radio, TV, information services and learning we can imagine a form of personalization that aims to provide to everyone the right information at the right time in the right place.

This dissertation deals with the topic of *adaptive hypermedia* and *adaptive Web-based systems*. We study this topic in general, and also look at the application of adaptive hypermedia in the context of *e-learning*.

In this introductory chapter we present the general research agenda. Section 1.1 describes the motivation of our research. Section 1.2 defines the research questions and approaches. Section 1.3 presents an outline of this dissertation.

1.1 Motivation, Background and History

This research brings together two disciplines — **Cognitive/Learning Styles**, a research topic in *cognitive psychology*, and **Adaptive Hypermedia**, a *computer science* topic.

We will start by motivating our choice for the field of Adaptive Hypermedia (AH) [Brusilovsky, 1996, 2001], and this will almost automatically lead to the second topic, of Cognitive/Learning Styles (CS/LS).

1.1.1 Adaptive hypermedia

Adaptive Hypermedia deals with the topic of *adaptation* and the topic of *hypermedia*. *Hypermedia* is a research field with a long history, starting with the design of the (never implemented) *Memex* system [Bush, 1945] in 1945, and the invention of the term *hypertext* by Ted Nelson around 1965. Hypertext is non-sequential text. Pages, or *nodes* are connected with *links*. Since the introduction of World Wide Web, hypertext has become the de facto standard way to communicate information. We also use the term *hypermedia*, to indicate that the information is not only non-sequential but also *multimedia*, meaning a combination of different media (such as text, images and video). Throughout this dissertation we will use the terms hypertext and hypermedia interchangeably.

Using hypertext users can jump from one page to another by following a link between them. This (repeated) action is often called *browsing*, and the user interface commonly used for this is called the *browser*. If there are many links, the user will have a lot of navigational freedom. However this leads to certain difficulties. The major usability problem in hypermedia is usually seen as the *disorientation* many people experience [Conklin, 1987]. It is also known as the "lost in hyperspace" problem. Books are often read in a more or less linear fashion, so it is easy to know where you are and what you have read before. In a hyperspace, especially when it becomes huge, there is a high probability that the user gets lost and does not know how the "current" page fits into the big picture. When there are many links it is also difficult for the user to choose which link to follow. This causes "the additional effort and concentration necessary to maintain several tasks or trails at one time" defined as *cognitive overhead* [Conklin, 1987]. This problem is often also referred to as "information overload". To tackle these problems, hypermedia systems could *guide* users through the information towards the pages that contain relevant information (related to the users' interest), and they need to provide context information so the user knows the position of the information in the entire hyperspace.

Besides different interests, users also have a different background and level of knowledge in particular domains. The same material may be understood differently by different people. It can be rather easy for a user with some preliminary knowledge of the subject domain and difficult for a user without any basic knowledge. In a (linear) book this is not much of a problem because the author knows in which order the reader visits the pages, but in hypertext it is impossible to foresee all possible ways in which users visit pages, and thus impossible to foresee what the user has read before visiting a certain page. We therefore introduce the term *personalization* which is the process of presenting information in a way suitable for each individual user. A personalized information source should present different information to different users. (Normal) hypertext does not really provide personalized information. The hypertext system may offer links to different versions of the information, however it cannot automatically guide the user to the version that is most suited for him or her.

A new generation of hypermedia systems — *Adaptive Hypermedia Systems (AHS)* — appeared with the purpose of solving these problems. Adaptive Hypermedia research originated in the early 1990s from the coming together of the areas of *hypermedia* and *user modeling*. User Modeling is the process of building up knowledge about a user in an effort to form a representation of his/her mental state. The field of user modeling provides techniques to collect user information, maintain user profiles (or user models) and deploy this information to provide personalized information to users.

Adaptation refers to the fact that the application changes (adapts) its behavior for each individual user. Adaptation can be based on various user features stored in the user model (UM), and also on the characteristics of the user environment like the user's location, time, the device used for interaction, etc.. In this dissertation we concentrate on the adaptation to the user's personal characteristics as stored and maintained in a user model (both domain specific information such as the user's knowledge about the application domain and domain independent information such as the user's cognitive abilities).

An adaptive application can change the information it shows. For example, it can provide an expert in a certain domain with more detailed and specialized information than it would provide to a novice. It can change different aspects of this information, like the media used, the length of the presentation, the level of difficulty, style, etc., all depending on the user's capabilities and preferences. Also links offered to the user and the presentation of these links can be changed. The user is guided towards relevant information so that (s)he can reach interesting information more easily and quickly. At the same time AH does not restrict users, keeping the navigation freedom of traditional hypertext. It shows the relevance of links but lets users move freely through the pages of an application (even if the links are not relevant). Traditionally in the field of adaptive hypermedia these forms of adaptation are distinguished as *adaptive presentation* (or content adaptation) and *adaptive navigation support* (or link adaptation) [Brusilovsky, 1996].

AHS have been under development since the 1990's in various application areas. Many systems were developed since that time offering mostly similar features and in some case reinventing the same ideas. The systems were described in different ways and often had one or more new ideas or features (among many previously invented ones). It was generally difficult for people to understand and compare different AHS. With the purpose of providing a single reference model for existing AHS the Adaptive Hypermedia Application Model (AHAM) was developed in 1999 at the Eindhoven University of Technology [De Bra et al., 1999; Wu, 2002].

The most popular application area for adaptive hypermedia research is *adaptive educational hypermedia systems (AEHS)*. The goal of the user (learner) is usually to learn all the learning material or a reasonably large part of it. The most important feature in this area is the user's knowledge of the subject area being studied. Every AEHS is able to perform adaptation to this knowledge. Other learner features the AEHS may take into account are background, hyperspace experience, preferences and interests. These properties tend to relate to the application domain. Less attention has been paid in AH to the fact that people have different approaches to learning, namely that individuals perceive and process information in very different ways. We refer to these differences as *Cognitive/Learning Styles*. Our goal is to bridge this gap. We wish to develop an adaptive hypermedia platform that is capable of performing adaptation to many different aspects. Adaptation to knowledge, background, preferences and interest will of course be part of it, but adaptation to the learning styles will be used to show that the "adaptive power" of the developed platform is greater than that of many other (special purpose) AHS.

1.1.2 Cognitive/learning styles

The nature of cognitive and learning styles is studied by cognitive psychology. There is considerable confusion in the literature concerning both terms. Various definitions of cognitive and learning styles are provided by different authors, but no universally recognized definition has been identified. According to some psychologists most definitions of learning style as well as cognitive style illustrate variations in individual information processing [Heineman, 1995]. Cognitive style deals with the "form" of cognitive activity (i. e., thinking, perceiving, remembering), not its content. Cognitive styles have been studied extensively in academic research [Liu and Ginther, 1999]. Learning style, on the other hand, is seen as a broader construct, which includes cognitive along with affective and psychological styles [Keefe, 1979]. Learning styles have been studied mostly in conjunction with practical applications [Liu and Ginther, 1999]. A major difference between the terms as used in literature is the number of style elements involved. Specifically, cognitive styles are more related to a bipolar dimension while learning styles are not necessarily either/or extremes. Numerous authors use the terms interchangeably [Heineman, 1995; Liu and Ginther, 1999]. Throughout the dissertation we often use just one term "learning style(s)", or LS for short. We refer to the definition of learning styles accepted by the leading theorists "as relatively stable indicators of how a learner perceives, interacts with and responds to the learning environment" [Keefe, 1979].

LS can indicate a user's preferences for different types of information or different ways of navigating through or interacting with the information space. For example, users with a *visual* style prefer to get more pictorial information, whereas about the same topic users with a *verbal* style prefer to get a more textual description (which can be provided through written text or spoken audio). Moreover, some learners prefer to learn things by doing something actively first whereas others prefer to collect data first and then turn to action. This corresponds to the *active* and *reflective* learning styles, respectively. Some learners tend to learn through a linear, step-by-step process which is logical and systematic, whereas others want to see the big picture before they tackle the details. This corresponds to the *sequential* and *global* learning styles respectively. (We refer here to the learning styles of the Felder-Silverman learning styles model [Felder and Silverman, 1988].)

Experiments in educational settings show that matching/mismatching a user's LS with the design of an instruction can be an important factor with regard to learning outcome. A number of studies indicate that the user's performance is much better if the teaching methods are matched to the user's LS [Chen and Macredie, 2002]. At the same time, other experiments show that for more able users mismatching learning materials to LS may be advantageous as it encourages users to develop learning strategies that could cope with a wider range of materials and experiences in the future [Holodnaya, 2002]. There are also a number of studies that do not show any significant difference between the learning outcomes while matching/mismatching design of an instruction with LS [Hayes and Allison, 1993;

Coffield et al., 2004]. Despite this we consider that it is worthwhile that learners have the choice to ask the AHS to adapt the learning material to their LS. Hence, for our research it is also important that the system can "analyze" the user's usage patterns in order to "detect" what the user's LS is.

1.1.3 Connection between cognitive/learning styles and adaptive hypermedia

Recently, several computer science researchers have been trying to integrate LS in the design of their adaptive applications (see below). Nevertheless, this is not an easy process. One of the difficulties in designing hypermedia software that incorporates LS is their representation in such an environment. The literature reveals that there have been very few studies that have set out specifically to investigate the relationship between LS and hypermedia applications, especially adaptive versions [Clarke, 1993; Daniels, 1996].

A few systems have been developed with regard to various LS — AES-CS [Triantafillou et al., 2002], APeLS [Canavan, 2004], CAMELEON [Laroussi and Benahmed, 1998], CS388 [Carver et al., 1996], INSPIRE [Grigoriadou et al., 2001], iWeaver [Wolf, 2002], MANIC [Stern and Woolf, 2000], Tangow [Carro et al., 1999]. For each of these systems the developers selected a number of LS and implemented support for them by defining corresponding instructional strategies. Therefore the choice of LS and definition of the strategy is done by the developers of the system rather than by the authors (designers) of the applications. In our opinion this approach limits the abilities of authors to create LS support. We think that developers should provide enough flexibility for the authors to define their own strategies and decide which strategies to apply for a particular application. Furthermore, different authors can have different visions on the same instructional strategy — they may expect different types of adaptation for a user with a particular LS.

In most of the systems the LS are assessed through psychological questionnaires and psychometric tests. These questionnaires are quite long because determining learning characteristics is very difficult. However, they are not always reliable and valid [Holodnaya, 2002; Coffield et al., 2004]. Furthermore not all characteristics they measure are stable and invariable across different subject domains [Holodnaya, 2002; Coffield et al., 2004]. The time consuming process of filling out questionnaires which might also give erroneous results is often undesirable.

In our opinion it would be useful to enrich the systems with some mechanisms to unobtrusively detect the learner's preferences, which might indicate a certain LS. This could be done by observing how the user interacts with an application. In this way the existing psychological questionnaires cannot be replaced completely, however some repetitive patterns in the user's browsing behavior can be observed. As a result, through the simplified and unobtrusive mechanisms, the dynamic adaptation of the system to identified user preferences can be provided.

Therefore the main purposes of our research on LS adaptation can be seen from two perspectives:

- *End-users of the applications*: Our goal is to provide end-users with different instructional strategies for working with the application. These strategies should be created by the authors of adaptive applications and associated with various LS, though they might be not limited to them. There should be an option to let the user switch between strategies, try different ones and choose the preferred strategy. Another goal is to provide a mechanism to infer some user characteristics which might correspond to a particular LS.

- *Authors of adaptive applications*: As we are not psychologists, and have not performed research on LS, we do not feel qualified to recommend any particular instructional strategy for a particular LS. We look at the strategies that are recommended by the cognitive science literature and show how to "translate" them to adaptation in AHS. Our aim is to allow the authors to design their own strategies from scratch or to create variations of existing strategies and define how to apply the strategy/strategies (in which combination and order) to their applications. In this way the authors can be free to experiment with their strategies and see which of the variations can represent certain LS in the best way. Creating an adaptive application with support for LS requires a lot of psychological knowledge - how to structure the application, how to provide content alternatives to present equivalent information for users with different LS. As the developers of the system, we do not wish to make these decisions but only provide an authoring environment so that authors can design such an application.

This dissertation analyzes the possibilities of providing support for adaptation to LS in AHS in general and in the AHA! (Adaptive Hypermedia Architecture) system in particular. AHA! is a Web-based adaptive hypermedia system intended to serve many different purposes. It was first developed as an educational system at the Eindhoven University of Technology (TU/e), around 1996. In the initial stages of this dissertation work AHA! was turned into a *general-purpose* tool aiming at bringing adaptivity to a wide variety of applications such as on-line information systems, on-line help systems, museum and shopping websites, in addition to the area of on-line textbooks.

We show the implementation of our ideas on LS adaptation in the AHA! system. This has been done in two steps:

1. At the start of this research work AHA! version 1.0 was available. At that time we were not (yet) thinking of incorporating LS in AHA!, but started working on turning AHA! into a general-purpose tool. Using AHA! 1.0 it would not have been possible to implement strategies for LS. Several people, including the author of this dissertation, have been working on AHA! to enrich it with the features described in the AHAM reference model [De Bra et al., 1999; Wu, 2002]. A number of new features allowed us to start thinking about new AHA! extensions, including providing adaptation to some higher-level traits, rather than just properties of the subject domain. To show the support for these higher-level traits we have chosen LS.

2. The new AHA! (version 3.0) features are rich enough to allow us to implement a number of strategies for LS. However the authoring process for LS adaptation, using

standard AHA! 3.0, would be far too complex for even sophisticated authors. Therefore the next step was to extend the AHA! authoring environment with authoring tools to provide adaptation to different LS, and also to provide designers with the ability to *define* different strategies for various styles while creating the adaptive applications. In order to show the general-purpose nature of AHA! we have provided the LS support "on top" of the (unmodified) AHA! 3.0 system, meaning that this extension does not affect the existing AHA! engine for content delivery.

1.2 Research Questions and Approaches

This dissertation addresses the following research questions:

Research question 1: Is an AHS necessarily a special-purpose tool?
When we look into the history of adaptive hypermedia we see that AHS have been developed to serve a specific application or application area. As a result the systems are all *special-purpose* tools. However, when we consider the AHAM reference model that tries to capture the functionality of existing (and perhaps some future) AHS it appears that a system that comes close to this reference model might be possible to build. And since such a system would have the functionality of (many) different existing systems it would indeed be a *general-purpose* tool. In this dissertation we first formulate the design requirements a general-purpose tool should satisfy and then we answer this research question by designing and building a system that is intended as a *general-purpose* or *multi-purpose* AHS.

Rather than start from scratch (or from just the AHAM model) we have developed the general-purpose AHA! system (version 3.0) based on the adaptive engine used in the course "Hypermedia Structures and Systems", known as 2L670 and later 2L690, at the Eindhoven University of Technology. This engine was a typical educational adaptive hypermedia engine, performing adaptation to the learners' knowledge about the different concepts that are described in the course.

In order to prove that we indeed created a general-purpose tool we could have built with it a lot of examples targeting various application areas. However we considered that this would be equivalent to being "Jack of all trades, master of none" (a term describing someone being competent in many endeavors but excelling in none). We therefore have chosen a different approach and decided to concentrate on one particular application area. In order to consider whether a completely different kind of adaptation is possible in our attempted general-purpose system we have considered the issue of *learning styles* to be applied in educational applications. We consider two research questions regarding learning styles:

Research question 2: Can we specify how an AHS should perform adaptation according to different learning styles?
To make a connection between the LS and their representation in adaptive hypermedia we analyze the recommendations from psychological and computer science research. We investigate how other researchers tried to incorporate the LS into their AHS. We want

to find out if there is a single and "correct way" to represent a particular LS in AH. We discuss the limitations of the systems trying to solve this problem and propose our view upon it.

Next, we want to incorporate support for LS in the AHA! system. Therefore the next question is:

Research question 3: Can the adaptation that is required for LS be realized through the AHA! system?
This research question boils down to two different issues, because there is a difference between what the AHA! adaptation engine can do and what the AHA! authoring tools support.

Research question 3a: Can the AHA! engine perform the adaptation that is required for different LS?

The answer to research question 2 will tell us what the adaptation is that is needed to suit different learning styles. To answer research question 3a we need to show the AHA! adaptation rules and other aspects of the presentation generation in order to demonstrate that the adaptation to different LS is possible. This shows the ability to perform the adaptation, but leaves the question whether authors will be able to define that adaptation.

Research question 3b: Can the adaptation to LS be created using the existing ("standard") AHA! authoring tools?

The basic principle of the AHA! authoring process is that high-level constructs are created by a designer, and are then used or applied by an author. Therefore the research question boils down to the ability of the system developers to create the required high-level constructs and their translation to the low level adaptation rules. AHA! 3.0 comes with a high-level authoring tool "Graph Author" for creating adaptive applications, based on *concepts* and *concept relationships*. We show how we can represent LS using only the features of the Graph Author. We then explain the necessity for extending the AHA! authoring system in order to allow authoring of the strategies at a higher level and discuss a proposed language for strategies description. We analyze different types of strategies we want to be able to incorporate in AHA!, and how they should be visualized in adaptive applications.

1.3 Outline of the Dissertation

This dissertation consists of seven chapters.

In chapter 1 we outlined the main research questions and hypotheses.

Chapter 2 gives an overview of adaptive hypermedia systems and existing models for adaptive hypermedia focusing on the Adaptive Hypermedia Application Model (AHAM). It starts addressing research question 1 by providing a list of design-requirements for a general-purpose AHS.

Chapter 3 provides an answer to research question 1 by presenting the design and implementation of such a tool. It presents the overall AHA! 3.0 architecture and different AHA! sub-models in relation to the AHAM model. Furthermore, the chapter pays attention to the issue of authoring adaptive applications in AHA!.

Chapter 4 presents general information about LS and emphasizes those LS that, according to psychological researchers, have possible implications for pedagogy.

Chapter 5 answers research question 2. It discusses the suggestions from psychological and computer science research for adaptation of instruction design to a number of LS. It presents examples of existing AHS providing adaptation to a number of LS.

Chapter 6 answers research questions 3, 3a and 3b. It presents our approach to incorporating learning styles in AHA! 3.0. It compares the presented approach with the approaches adopted in other systems providing adaptation to LS. It also shows the results of evaluation of this approach with the students of Eindhoven University of Technology and the results of approach validation performed by a number of learning styles experts.

Finally in chapter 7 we give a summary of the main results and indicate some directions for future research.

Chapter 2

Adaptive Hypermedia Systems (AHS)

This chapter provides background information on the adaptive hypermedia field, as needed to understand the research presented in this dissertation. From this information we derive the *design requirements* for creating general-purpose adaptive hypermedia systems.

Section 2.1 introduces the research field of AHS. Section 2.2 reviews their application areas. Section 2.3 discusses the issues related to modeling users in AHS. Section 2.4 outlines methods and techniques for providing adaptation.

Although developed for different application domains many AHS offer very similar features and functionalities. The same ideas are repeatedly being reinvented. In order to provide a standard reference architecture for AHS the Adaptive Hypermedia Application Model (AHAM) was developed in 1999 at the Eindhoven University of Technology. AHAM and a few other models for adaptive hypermedia that were developed later are outlined in section 2.5.

Section 2.6 presents several examples of AHS. Section 2.7 discusses the issues related to the creation of adaptive hypermedia applications (rather than systems).

Most of the existing AHS were developed for applications in one specific area, for instance, in educational settings. As a result they cannot be reused in other domains, such as e-commerce or information retrieval. In section 2.8 we start addressing research question 1 of this dissertation by formulating a list of design requirements for implementing a general-purpose AHS, a tool that would allow the creation and delivery of adaptive applications targeting various areas. We base our research on the application-independent architecture proposed by the AHAM model. We provide an answer to research question 1 in chapter 3 by actually designing and implementing such a general-purpose system.

2.1 Introduction

The development of adaptive hypermedia systems started in the early 1990s on the cross-road of two areas — Hypertext and User Modeling. The first research efforts to explore various ways of adapting the output and behavior of hypertext systems to individual users were independent and the early researchers were generally not aware of each other's work.

The support from the already established user modeling research community was influential in helping the existing research teams to find each other, and in recognizing and promoting adaptive hypermedia as an independent research direction [Brusilovsky, 2001]. The first overview of AHS, methods and techniques used by these AHS was given by Peter Brusilovsky in 1996 [Brusilovsky, 1996].

The year of 1996 marks a turning point in adaptive hypermedia research [Brusilovsky, 2001]. Before this time research in the area of adaptive hypermedia was performed by a few isolated teams and most of the developed systems were not Web-based. After 1996 the field saw an expansion with the development of a broader range of AHS, techniques, fast adoption of Web technology and the first official conference in 2000. Brusilovsky gave a new version of the overview of adaptive hypermedia systems in 2001 [Brusilovsky, 2001].

Objectives of AHS

The goal of adaptive hypermedia research is to improve the usability of hypermedia applications by making them *personalized*. AHS can be useful in any application area where users of a hypermedia system have essentially different goals and knowledge and where the hyperspace is reasonably large. AHS provide a certain level of intelligence to hypermedia systems in the sense that they have the ability to "understand" the user and to adapt their behavior to the user's needs. Having knowledge about the users, AHS can support them in navigation by limiting the browsing space, suggesting the most relevant links to follow, resulting in decreasing search and navigation time; they also support the improvement of comprehension of the content by presenting the most relevant information on a page and hiding information that is not relevant. Their aim is thus to solve the "lost in hyperspace" and "information overload" problems.

These objectives are achieved by collecting information about the users while they interact with the system, and by adapting the application based on the gathered information. This information is stored in the so-called *user model*, also referred to as a *user profile*. (We describe our distinction between these two terms below.)

Definition of AHS

Peter Brusilovsky provided the following definition of AHS [Brusilovsky, 1996]:

> "By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system; it should have a user model; it should be able to adapt the hypermedia using this model."

A clear distinction has to be made between *adaptable* (also called *customizable*) and *adaptive* hypermedia which together comprise personalized hypermedia. In both cases the user plays a central role and the ultimate goal is to offer a personalized system. The difference is in the way the adaptation is performed:

- An *adaptable* hypermedia system allows the user to configure the system by explicitly setting his/her features in a *user profile* (for instance, through a form) and the system adapts its behavior accordingly. Adaptable hypermedia systems do not change the user profile unless the user explicitly updates it.

- An *adaptive* hypermedia system constructs and maintains a *user model* by observing his/her behavior (most often browsing behavior) and adapts automatically to the current state of that user model. An AHS thus works "in the background", without asking the user for specific input on his/her goals, preferences or knowledge.

Many of the existing systems are both adaptable and adaptive. The *user profile* is then a part of the *user model*. In the case of adaptive systems we will not user the term "user profile" but only "user model", to refer to the static and dynamic parts together.

Another class of systems are the *dynamic* hypermedia systems [Cannataro and Pugliese, 2004]. They do not have predefined presentations, the system generates presentations "on the fly" by combining atomic information fragments in a completely dynamic way, on the basis of the user model.

Problems and difficulties of AHS

Although AHS bring many benefits there are several problems related to their usage. Here we mention some of them.

1. *"Unstable" interface.* The behavior of an adaptive application might be confusing for the user. It relates to "instability" of the presented pages. For example, if the user returns to some pages, they may look different from how they were presented before, as they are generated according to the current state of the user model. The user might become irritated that (s)he cannot find some information or links (s)he has seen before on these pages. Several systems are trying to eliminate this risk. SmexWeb [Albrecht et al., 2000] keeps a history of changes to the user model. It allows pages to have the same look and feel throughout a session for a particular user. The AHA! system (described in detail in this dissertation) also provides solutions for "stabilizing" presentations by keeping concept representations stable [De Bra et al., 2003a]. The stability can be chosen to be forever after the first adaptation, stability during the current session, or stability while a certain Boolean expression remains true (an expression which uses the values of the attributes in the user model).

2. *Privacy issues.* The concept of *personalization* is subject of ethical (and legal) debate. People fear misuse of personal information gathered by the systems. Some legal issues that have to be considered in adaptive hypermedia are summarized in [De Bra et al., 2004a]:

- Users should have a choice between an "anonymous" login (or login using a pseudonym) and a normal login using their real identity. They should be also allowed to change or erase data, in particular the assumptions the system inferred about them.
- Users should be able to inspect their user model and understand what it means.
- The user models should be protected using adequate security measures from intruders. This refers to the situation when user modeling becomes a service[1], accessible by multiple applications.

An in-depth review of privacy concerns in adaptive hypermedia can be found in [Kobsa, 2002].

3. *Difficulty of the authoring process.* Despite the benefits adaptive hypermedia can offer it is still not widely used. This might be because authoring of adaptive hypermedia is a very complex task, good authoring tools are lacking, reuse of authored materials is difficult, the structural overview is lacking, the adaptation is hidden in the implementation, authoring cooperation is almost impossible and finally there is no standardized approach to adaptive techniques and behaviors [Cristea and Cristea, 2004]. Furthermore, people with very different skills are involved in the process, such as authors, layout designers, programmers, multimedia experts and also marketing specialists of e-commerce applications. Currently a number of researchers are trying to attract more attention to this problem [Stewart et al., 2004; Kravcik et al., 2004; De Bra et al., 2003b]. A number of approaches to authoring adaptive hypermedia are described in section 2.7.

2.1.1 Identified dimensions for adaptation

The following dimensions for analysis of AHS are identified in [Dieterich et al., 1993]:

1. *Where can AHS be useful?* Section 2.2 reviews several application areas for AHS.
2. *What to adapt to?* This dimension identifies users' characteristics which can be used for adaptation. The issues related to user modeling in AHS are presented in section 2.3.
3. *What can be adapted?* This dimension refers to features of AHS which can be adapted and different ways to adapt hypermedia, called by Brusilovsky [Brusilovsky, 1996, 2001] technologies of adaptation:
4. *Adaptation goals* achieved by different adaptation methods and techniques. In section 2.4 the adaptation goals are considered in parallel with the review of relevant adaptation methods and the techniques that implement these methods.

[1] According to World Wide Web Consortium definition (http://www.w3.org/TR/ws-gloss/#webservice): "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network."

These sections help us to derive a number of requirements for providing a general-purpose tool.

2.2 Application Areas of AHS

Brusilovsky classified AHS into six groups according to their application areas:

- *Educational hypermedia systems* (examples InterBook [Brusilovsky et al., 1998], KBS Hyperbook [Nejdl and Wolpers, 1999]) — this is the most popular area for adaptive hypermedia research. The most important user feature in this area is the user's knowledge of the subject being studied.

- *On-line information systems* and on-line documentation and encyclopedias (e. g., Swan [Garlatti et al., 1999], MetaDoc [Boyle and Encarnacao, 1994], GUIDE [Cheverst et al., 2002], WebGuide [Fink and Kobsa, 2002], Intrigue [Ardissono et al., 2003]) — the goal of these systems is to provide reference access to information in a hyperspace. The information space in these systems can range from small to very large. This group includes electronic encyclopedias, information kiosks, virtual museums, handheld guides, e-commerce systems, performance support systems. Applications of this group differ from educational AHS as they do not present a systematic introduction to a learning subject.

- *On-line help systems* (e. g., ORIMUHS [Encarnacao, 1997], I-Help [Vassileva et al., 2001]) — in some ways they are also on-line information systems, but the objective is to assist the user when (s)he has difficulties with the tool or system. The assistance consists of presenting help information when requested (as in on-line information systems) and automatically recognizing when the user needs some help. Adaptive on-line help systems have the advantage of knowing the context in which the user is working, therefore they allow context-sensitive help. Unlike on-line information systems, these type of systems are always attached to tools or other systems. They are not independent from their application systems and their information space is reasonably small.

- *Information retrieval (IR) hypermedia systems* (e. g., WebWatcher [Mladenic, 1999], ifWeb [Asnicar and Tasso, 1997]) — combine traditional information retrieval systems with hypermedia features. The systems provide the possibility of browsing the information space using similarity links between documents. The objective is to obtain a manageable set of responses to a query put to an information hyperspace as opposed to a query database. The responses are a set of links calculated by the search engine. These systems typically limit the navigation choice and give hints as to which are the most relevant links.

- *Institutional information systems* (e. g., Hynecosum [Vassileva, 1996]) — serve on-line all the information required to support the work of some institution, for example, a hospital. It is work-oriented in the sense that most users only need to access a specific area of the information space relevant to their work. This is significantly different from search-oriented IR hypermedia and on-line information systems where the "working area" of a user is the entire information space.

- *Personalized views* (e. g., PowerBookmarks [Li et al., 1999], WebTagger [Keller et al., 1997]) — are developed to manage a huge and dynamically changing hyperspace in a personalized way. By defining their personalized views on the entire hyperspace, users protect themselves from the complexity of the overall hyperspace. Each view can be devoted to one of the goals, background or interests related to the work of the user. Similar to institutional hypermedia, users of these systems need convenient access to a subset of an information space for everyday work. Personalized views in world-wide information spaces require permanent management because of the dynamic character of the information space.

2.3 Modeling the User

Information which is stored in the user model provides a basis for adaptive behavior of an AHS. What aspects of the user working with the system should be taken into account when providing adaptation?

Kobsa et al. [Kobsa et al., 2001] distinguished three categories of user related data:

- *user data* — refers to information about personal/individual characteristics of the user. Brusilovsky [Brusilovsky, 2001] identified a number of user features which AHS can store in the user model and apply for adaptation: knowledge, goals/tasks, background, hyperspace experience, preferences, interests and individual traits (e. g., cognitive/learning styles).

- *usage data* — is related to information on the user's interactive behavior that cannot be resolved to user characteristics. Examples of types of interactions are: selective actions (e. g., clicking on a link, scrolling and enlarging operations for hypermedia objects, audio control operations), temporal viewing behavior rating (users are required to explicitly rate objects, links, webpages), purchases and purchase-related actions, etc.. These interactions can be observed directly and used immediately in the adaptation process. In many cases some observable user interactions cannot directly lead to adaptation, further processing of the usage data being necessary. Examples of usage data that is acquired after the information was processed are: usage frequency, action sequences and situation-action correlations.

- *environment data* — comprises aspects of the user environment that are not related to the users themselves. This kind of adaptation has evolved due to Web-based systems. Adaptation decisions may depend on spatio-temporal location of the user, the user platform (such as hardware, software, equipment — computer, PDA, cellphone, digital TV), network.

Depending on the level of granularity [Collins et al., 1997] (or the level of detail of the issues represented by the model) some user models are described as *fine-grained* or *coarse-grained*.

There is also a differentiation between short-term and long-term user models [Rich, 1999]. Short-term models are available as long as a session lasts. Long-term models are kept between sessions, perhaps even indefinitely.

Furthermore user models are generally distinguished into two main categories [Cannataro and Pugliese, 2004]:

- *Overlay models*. In an overlay model a user is described through a set of *concepts* with *attribute-value* pairs. For every concept and attribute in the domain model of the application a concept and attribute is kept in the user model. "knowledge" is a typical attribute of the concepts in educational applications.

- *Stereotype models*. Stereotype models indicate that the user belongs to a group of users having common characteristics. A distinction is made between *pure stereotypes*, indicating only that the user belongs to a group, without giving characteristics of the group, *mixed stereotypes*, which point out group characteristics through attributes, and *multiple stereotypes* in which a user can be associated with different groups at the same time. The user assigned to a stereotype inherits all the properties of that type. Stereotype modeling is reasonably reliable and works well for the systems that need to adapt to different classes of users (but not to specific characteristics of individual users). Good results are obtained by the combination of stereotyping techniques and overlay modeling. The initialization of the user model is then done by assigning a (mixed) stereotype to the user. The user model is then refined at each interaction/navigation step, thereby implementing an overlay model. An example is the user model implemented in ARCADE [Encarnacao and Stork]. Stereotype models are enough when only modeling the interface or choosing the type of instruction. They are insufficient when individual adaptation requires a more fine-grained description of the user or specific help or advice is required.

2.4 Methods and Techniques Used in AHS

Following Brusilovsky [Brusilovsky, 1996, 2001] we distinguish between high level methods for adaptive hypermedia support and lower level techniques that are used to realize or implement that support. A *method* is defined as a notion of adaptation that can be presented at the conceptual level. A *technique* is then a way to implement a specific method. Techniques operate on actual information content and on the presentation of hypertext links. It may be possible to implement the same method through different techniques and to use the same technique for different methods.

In 1996 Brusilovsky developed a taxonomy providing a way to segment adaptive research into two categories, which he labeled *adaptive presentation* and *adaptive navigation* *support*. Figure 2.1 shows an updated taxonomy developed in 2001 in the light of more recent research.

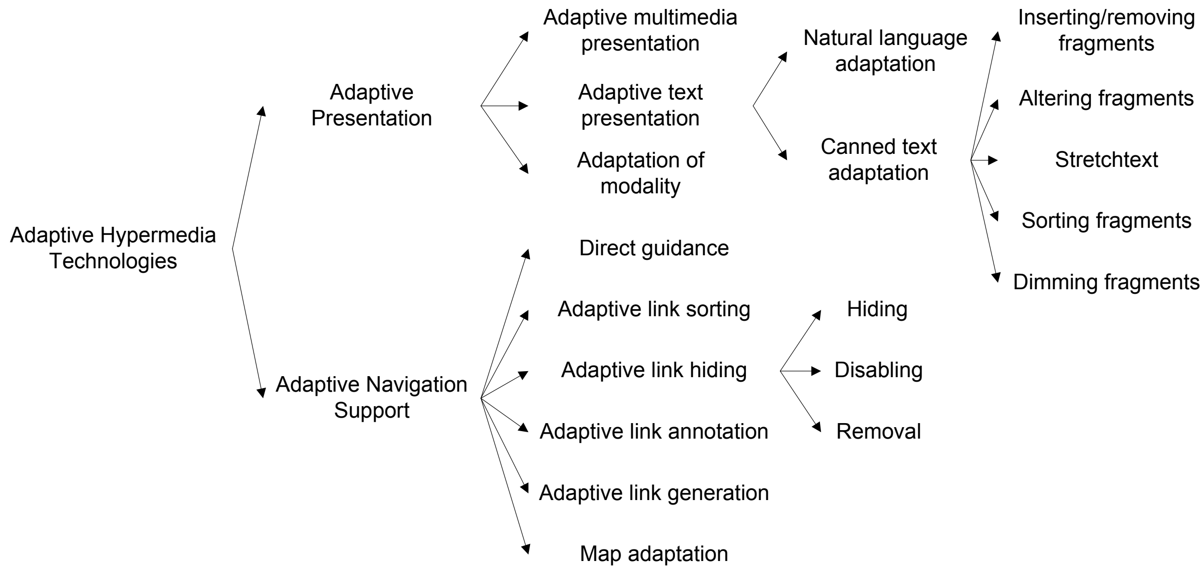


Figure 2.1: The taxonomy of adaptive hypermedia technologies [Brusilovsky, 2001]

A detailed overview of the taxonomy can be found in [Brusilovsky, 2001]. In this dissertation we only outline the most widely used methods and techniques for providing adaptation.

2.4.1 Adaptive presentation

The goal of adaptive presentation is to adapt the content of a hypermedia page to the user's goals, knowledge and other information stored in the user model. This area concerns the selection and composition of information fragments to be later presented to the user. For example, the system may decide to provide additional explanations as well as to hide content from a novice user, while an expert user in the domain will receive more specific and more detailed information. In a hypermedia system the content of a "page" may not only be text, but it may also contain various items of other media types. The main benefit of the adaptive presentation is that it tries to reduce the amount of presented information to the most relevant information for a particular user, solving the "information overload" problem of the classic hypermedia systems. In [Brusilovsky, 1996] three main adaptive presentation methods are considered:

- *additional, prerequisite* and *comparative explanations* (pieces of content that are sometimes shown) — this is the most frequently used method for adaptive content. The meaning is that under a given set of circumstances some additional content can be presented. Short prerequisite explanations may be added, comparisons to subjects described on other pages the user has seen before, additional details for advanced users, etc.. The most popular method appears to be the *additional explanations.* The goal of this method is to provide additional information, explanations, illustrations, examples, etc., to those users who appear to need or want them. Thus, apart from the basic presentation, additional information is provided for a certain category of users. At the same time the system hides such explanations from users who do not want them. The method *prerequisite explanations* is a special case of the first method. An explanation can be added because the system decides that without this explanation the user will (or may) not understand the remainder of the page. The method *comparative explanations* is used when there is information about other concepts that are related to the one described in the "current" page (similar or dissimilar in a specific way, or otherwise related in some "interesting way"). Showing this additional information about how the current topic relates to these other concepts only makes sense when these other concepts are known to the user.

- *explanation variants* — the meaning of this method refers to the fact that essentially the same information can be presented in different ways. Depending on values in the user model the level of difficulty, the related concepts a page refers to, the length of the presentation, the media type (text, images, audio, video) or other aspects may be changed. This can be done within a page or through guidance towards different page variants. (In the latter case the method becomes adaptive navigation support rather than adaptive presentation).

- *sorting* — the meaning of this method is that depending on user model values the order in which information items are presented may have to be altered. For instance, some users may prefer to see an example before a definition, whereas others prefer it the other way around. On a page, fragments of information are typically sorted from most to least relevant, a method which is best known from information retrieval systems.

Although there are various techniques of adaptive presentation the majority of work in this area has been categorized by Brusilovsky as *canned text adaptation*. This term includes fragment processing, such as:

- *inserting/removing fragments* — the name of this technique is equivalent to the term "conditional" text in the 1996 review. Information related to a concept is broken into several fragments of text (or multimedia content). With each fragment a condition is associated on elements of the user model. When information related to a concept is presented, the system selects only those fragments for which the condition is true. This technique can be used to implement the methods of additional, prerequisite and comparative explanations.

- *altering fragments* — this technique can be used for the implementation of the explanation variants method. The AHS stores several variants of the same information fragment, and selects the variant to display based on the user model. For example, each variant is created for different group of users such as beginner, intermediate and expert.

- *stretchtext* — a technique where fragments are embedded in a webpage and initially shown or hidden from the user depending on conditions on user model data. After the initial presentation the system will show or hide these fragments also upon explicit user requests. This technique is useful for implementing additional, prerequisite or comparative explanations, but less for explanation variants and not at all for sorting.

- *sorting fragments* — the idea of this technique is to present a set of fragments to the user, ordered from the most relevant to the least relevant according to some criteria based on user's goal, background, knowledge, etc.. This technique is used for implementation of the sorting method.

- *dimming fragments* — is a technique used to dim, shade or deemphasize (in some way) a fragment to indicate that it is not (or at least less) relevant for the user.

The other techniques are *adaptive multimedia presentation*: the selection of different multimedia fragments[2], and *adaptation of modality*: choice among different types of media to present to the user, related to the same content; e. g., for a video we can have full video, still image, text description, or use of them in parallel.

2.4.2 Adaptive navigation support

Adaptive navigation support modifies or augments the existing set of hyperlinks shown to the user to aid them in finding relevant information. The goal of adaptive navigation support is to help users to find their way through hyperspace by adapting link presentation and functionality to the goals, knowledge, and other characteristics of an individual user. The goal is to guide the user towards relevant and interesting information and to advise the user not to follow navigation paths that lead to irrelevant information. Whereas the adaptive presentation changes the content presented in the documents, the adaptive navigation support changes the structure of links between the documents and how this navigation structure is presented to the user. The main benefit of the adaptive navigation support is that it simplifies the rich link structure and solves the "lost in hyperspace" problem, while maintaining the navigation freedom that is typical of hypermedia systems. Brusilovsky [Brusilovsky, 1996] mentions five adaptive navigation support methods:

- *global guidance* — means that the system suggests navigation paths on a global scale, thus whole paths, not just single links;

[2] In principle this technique also includes the adaptation of multimedia content, just as the generation of shortened versions, but currently most examples of adaptive multimedia simply select between prefabricated content.

- *local guidance* — the system suggests the next step to take, for instance, through a "next" or "continue" button;
- *global orientation support* — the system presents a (filtered) overview of the whole (link) structure of the hyperspace;
- *local orientation support* — the system presents an overview of a small, local, part of (link) structure of the hyperspace;
- *managing personalized views* — each view may be a list of links to all pages or sub-parts of the whole hyperspace that are relevant for a particular working goal.

This branch of the adaptive taxonomy contains several different techniques that can be used individually or combined to provide navigational support:

- *direct guidance* — the system outlines visually one of the links on the page showing that this is the best link to follow or generates an additional dynamic link (often called "next" or "continue") which is connected to the recommended "next" page to visit.
- *adaptive link sorting* — the system sorts all the links of a particular page according to the user model and to some user-valuable criteria: the closer to the top, the more relevant the link is.
- *adaptive link hiding (hiding, disabling, and removal)* — the system tries to prevent the user from following links that are not relevant for him or her at the moment. There are several ways to achieve this. A link can be hidden by changing the color of the anchors to that of normal text. It can be disabled so that "link functionality" of a link will be removed — clicking on it will produce no effect. For undesirable links (non-relevant or not yet ready to be read) that appear in a list the link anchors can be removed. (In normal text you cannot do this because the sentences with missing words may no longer make sense.)
- *adaptive link annotation* — the system augments the links with some form of comment, which can tell the user more about relevance of the nodes behind the annotated links. These annotations are usually provided in the form of visual cues such as icons, font colors, sizes, and types.
- *adaptive link generation* — the system may discover new useful links between pages and add them, it may use previous navigation (by an individual user or a user group) or page similarity to add links. Generating a list of links is typical in information retrieval and filtering systems.
- *map adaptation* — regards the re-organization of the overall link structure of the hyperspace. This technique consists of a combination of the other techniques, the only difference being that it is applied to a graphical visualization of the navigation (link) structure. The map is usually presented in a separate frame or window.

2.4.3 More levels of adaptation

Another view upon the segmentation of the adaptation levels in adaptive hypermedia is presented in [Paterno and Mancini, 1999]. It distinguishes between changes to the layout that do not affect the content and provides the following classification:

- *adaptive content* — corresponds to adaptive presentation category in the Brusilovsky taxonomy;
- *adaptive navigation* — corresponds to adaptive navigation support category in the Brusilovsky taxonomy;

  a new level:

- *adaptive presentation* — shows different layouts of perceivable user interface elements, such as different colors, font size, font type or image size, but also provides multi-language adaptation (adaptation to the language preferred by the user).

2.5 Models for Adaptive Hypermedia

A wide range of AHS have been produced since the 1990s, however until 1999 there was no standard reference model or architecture for AHS. The *Adaptive Hypermedia Application Model (AHAM)* was the first one to be developed, later followed by the *Web Modeling Language (WebML)*, the *Munich reference model*, the *XML Adaptive Hypermedia Model (XAHM)* and *LAOS*.

The objectives of these references models are to capture important abstractions found in current adaptive hypermedia applications, to describe the basic concepts, such as the node/link structure of these systems, to provide a basis to compare the systems, and to develop a standard. The models describe, at a conceptual level, the components and descriptive languages used for specifying generalized adaptive hypermedia systems.

Here we provide an overview of these models. By analyzing their features we select a model to follow while designing a general-purpose AHS.

2.5.1 Adaptive Hypermedia Application Model (AHAM)

The development of AHAM, first introduced in [De Bra et al., 1999], was inspired by the work on the AHA! system. Since the development of AHAM, AHA! has been further modified to better fit into the AHAM model (see chapter 3).

AHAM is based on the Dexter model [Halasz and Schwartz, 1990, 1994], a formal reference model for hypertext systems developed in 1990. AHAM divides AHS into the same three layers as the Dexter reference model does for hypertext systems. As shown in figure 2.2 these are: the Run-time Layer, the Storage Layer and the Within-Component Layer, connected by the interfaces Presentation Specifications and Anchoring.

The focus of AHAM is the Storage Layer which has three parts (sub-models):

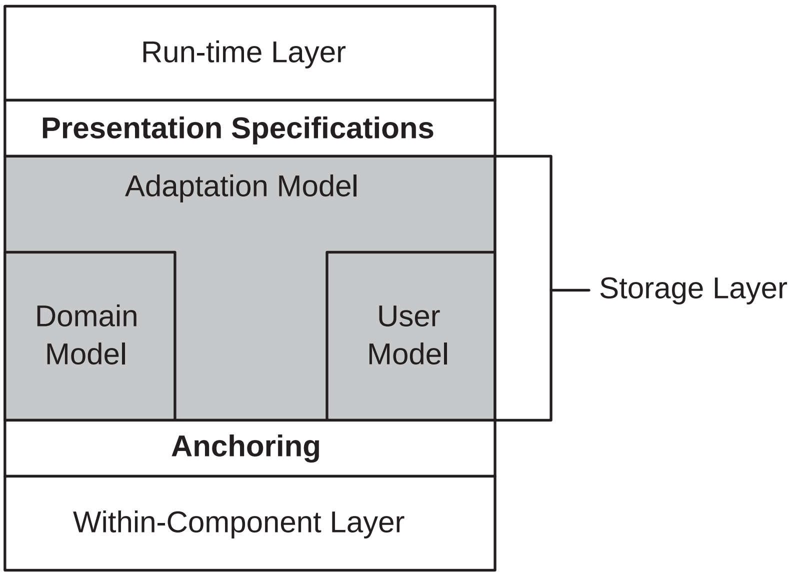


Figure 2.2: The AHAM model

- *a domain model (DM)* — describes the structure of the information content of the application. It consists of *concepts* and *concept relationships*. A concept is an abstract representation of an information item from the application domain. It can be either an *atomic concept* or *composite concept*. These types of concepts can represent fragments (atomic), pages and larger "units" of information. (Think of sections and chapters in a book.) All concepts have a unique id in the Storage Layer, but for the connection to actual content the (concrete) concepts are also uniquely addressed through the anchoring layer. Concepts have properties that are represented through *attribute-value pairs*. AHAM allows concepts to have arbitrary attributes of arbitrary data types. Each concept may have different attributes (although most applications normally have a lot of concepts with the same attribute set), and when two concepts have the same attribute it does not need to be of the same data type for each concept. A *concept relationship* is an object (with a unique identifier and attribute-value pairs) that relates a sequence of two or more concepts. Concept relationships can be directed or undirected. Binary directed relationships are most common (like normal hypertext links or prerequisite relationships) but other types are allowed too. In theory, AHAM authors or system designers can define arbitrarily many types of relationships.

- *a user model (UM)* — describes user features used in adaptation. The UM is an overlay model of the DM. For each concept in the DM the UM stores attribute values about the concept. The UM is fine-grained, based on the DM being well-structured using many different concept levels (from high level concepts down to fragments). It is possible to simulate personal characteristics and environmental factors through "pseudo concepts". Doing so enables AHAM also to describe AHS that take these other aspects into account, albeit in a somewhat artificial way. As with the DM, each concept in UM may have different attributes and the values may be of arbitrary data types.

- *an adaptation model (AM)* — provides the basis for adaptive functionality. The adaptation model contains a set of adaptation rules that are stated in the form of event-condition-action clauses. They use the structure and content of the domain model and the user model to decide how to update the user model and how to generate the adaptation. Therefore the AM is located between the DM and the UM in the Storage Layer.

  A distinction is made between *generic* and *specific* adaptation rules. In a generic adaptation rule, (bound) variables are used that represent concepts and concept relationships. A specific adaptation rule uses concrete concepts from DM or UM model instead of variables. Other than this, both types of rules look the same.

AHAM advocates a clear separation of the above three main parts of (the Storage Layer of) an adaptive application. This separation makes the role of each part and the relationships between the parts explicit. This also enables people with different skills develop an adaptive hypermedia application together by working on different parts separately. AHAM also advocates separating the Storage Layer from the more platform-specific aspects of the system to ensure platform-independence of the developed adaptive applications.

The AM generates Presentation Specifications. Presentation Specifications can indicate, for example, that content should be emphasized or that a link should be annotated in some way. These specifications are passed onto the Run-time Layer.

Everything relating to accessing and presenting the information to the user (the user interface) belongs to the Run-time Layer. It is up to the Run-time Layer to decide, possibly guided through style sheets, how to make these presentation aspects visible to the end-user.

All connections between the actual information content (in the Within-Component Layer) and the sub-models of the Storage Layer use connection points specified in the Anchoring Layer.

An adaptive hypermedia application consists of not only the structures defined by AHAM but also involves an adaptation engine (AE). AHAM describes the adaptation functionality at an abstract conceptual level; more specifically, it does not prescribe a specific algorithm for selecting and executing the adaptation rules. The AE describes the adaptation functionality at a (still abstract) implementation level. The AE executes the rules specified in the AM. It is thus an *interpreter* for the adaptation rules in the AM. When a user "clicks" on a link many adaptation rules can be executed. The AE chooses an order in which to execute these rules (or may choose to execute them in parallel). The rule execution may trigger other rules, for which the AE again has to choose an execution order. The AE thus plays an important role in the concrete behavior of an adaptive hypermedia application. Therefore an AHS is defined in AHAM as a 4-tuple:

$$AHS =< \textit{Domain model}, \textit{User model}, \textit{Adaptation model}, \textit{Adaptation engine} >$$

2.5.2 The Munich Reference Model

The Munich Reference Model [Koch and Wirsing, 2002] follows the same approach taken in AHAM by extending Dexter's storage layer with the user model and adaptation model. Architecturally the two reference models are almost identical.

The major difference between the two is that AHAM takes more a database point of view whereas the Munich Model takes an object-oriented software engineering one [Koch and Wirsing, 2002]. AHAM specifies an adaptation rule language, while the Munich model contains an object-oriented specification written in the Unified Modeling Language (UML)[3] which integrates both an intuitive visual representation and a formal unambiguous specification in Object Constraint Language (OCL, part of UML). Furthermore the AHAM adaptive engine is included in the adaptation model of the Munich reference model as data and functionality are integrated in the object-oriented approach [Koch and Wirsing, 2002].

2.5.3 LAOS

LAOS [Cristea and Mooij, 2003] — a generalized model for generic adaptive hypermedia authoring — is based on the AHAM model and concept maps [Novak and Canas, 2006]. LAOS defines five layers: domain model (DM), user model (UM), goal and constraints model (GM), adaptation model (AM) and presentation model (PM). There are a number of differences between AHAM and LAOS.

As can already be seen LAOS introduces a new layer which is the *goal and constraints* model. The goal and constraints model is taken from the book-presentation metaphor [Cristea and Mooij, 2003]: "generally speaking, when making a presentation, be it on the Web or not, we base this presentation on one or more references. Simplifying, a presentation is based on one or more books. With this in mind it is obvious why we cannot jump from the DM to the AM (or UM): it would be equivalent to skip the presentation and just tell the user to read the book. In other words, the search space is too big and there is a too high degree of generality (no purposeful orientation of the initial material — i. e., book). Therefore, what we need is an intermediate authoring step that is goal and constraints related: goals to give a focused presentation, and constraints to limit the space of the search." Within the educational domain, the goal and constraints model filters, regroups and restructures the domain model, with respect to an instructional goal [Cristea et al., 2005a]. The separation between DM and GM allows different pedagogues to apply various pedagogical strategies to the same domain model.

Another difference between AHAM and LAOS is the concept definition. The domain concepts in LAOS are semantically independent entities, defined by their attributes. This means that all necessary information for using a concept is provided in the concept itself,

[3] http://www.omg.org/uml

i. e., instead of connecting a concept to application page(s) the contents of a concept are specified within its attributes. Thus a story line can be generated by simply using the attributes of the concepts in LAOS.

The user model can be an overlay to either the goal and constraints or the domain model (e. g., the knowledge of a certain topic represented by a goal and constraints model concept). Alternatively the user model can also contain other variables not related to the DM and GM such as learning style, background, etc. [Cristea et al., 2005a].

The domain model, goal and constraints model, and user model are represented as concept maps in LAOS.

Furthermore LAOS has a composition of the adaptation model, different from that of AHAM. The adaptation model is based on LAG [Cristea and Calvi, 2003] — a three layer model for authoring adaptation. LAG, consists, at the lowest level, of an *Adaptation Assembly Language*, corresponding to the typical IF-THEN rules in adaptive hypermedia (IF <PREREQUISITE> THEN <ACTION>). At the intermediate level, the model requires a semantic *Adaptation Language*. This language should incorporate more semantics, which should allow reuse in different learning situations. Finally, at the highest level, the LAG model situates *Adaptive Strategies* or *Adaptive Procedures*. These strategies / procedures are containers for the actual adaptation program (which details, in a machine readable adaptation language, how the adaptation will be performed). In addition each strategy has a description (semantic label) in natural language, which can be directly used by authors to select a specific, ready-made strategy for their course. In this way, course contents creation and the creation of adaptation dynamics for that course are kept separate, and can be performed by the same or different authors, at different times. As an instantiation of the Adaptation Language in the LAG model, the LAG Language [Cristea and Cristea, 2004] was introduced. This language uses as its syntax the LAG grammar (discussed in more details in section 6.1.1 of chapter 6), and is the basis of an Intermediate Platform specification for adaptation dynamics. Concretely, the LAG Language provides the building blocks for the creation of Adaptation Strategies. Rather than dealing only with specific rules that use the concrete concepts the language allows for the definition of higher-level rules that use the variables representing domain model concepts and concept relationships. Furthermore the language provides a high level of semantics by allowing a number of alternatives to the conditional IF-THEN rules, such as level, temporal repetition rules, interruption, generalization and specialization commands [Calvi and Cristea, 2002].

Also the goal of the presentation model in LAOS is quite different from AHAM presentation specifications. [Cristea et al., 2005a] "Presentation has to take into account the physical properties and the environment of the presentation and provide the bridge to the actual code generation for the different platforms (e. g., HTML, SMIL, XHTML, etc.). Presentation makes the difference between different devices of display, such as handheld devices, desktops, laptops, etc.. This part in the LAOS model is concerned with the formatting so that the information appears nicely in the page, with questions such as the ideal page length, where chapters of the presentation should be cut to form pages, how and where multimedia should appear (from the point of view of display possibilities), colors, fonts, etc.."

2.5.4 WebML

WebML is a visual language for specifying the content structure of Web applications and the organization and presentation of contents into one or more hypertexts. Besides visual diagrams Web application design based on WebML can be represented as XML[4] documents as well [Ceri et al., 2007]. The specification of a website in WebML consists of four orthogonal perspectives [Ceri et al., 2000]:

- *Structural Model* — expresses the data content of the application, in terms of the relevant entities and relationships. WebML is compatible with classical notations like the E/R model [Chen, 1976], the ODMG object-oriented model [Cattell et al., 1997], and UML class diagrams [Booch et al., 1998].

- *Hypertext Model* — describes how contents, previously specified in the data schema, are published into the application hypertext. [Ceri et al., 2007]: "The overall structure of the hypertext is defined in terms of *site views*, *areas*, *pages* and *content units*.

  A *site view* is a hypertext, designed to address a specific set of requirements. Several site views can be defined on top of the same data schema, for serving the needs of different user communities, or for arranging the composition of pages to meet the requirements of different access devices like PDAs, smart phones, and similar appliances. A site view is composed of *areas*, which are the main sections of the hypertext, and comprise recursively other sub-areas or *pages*. Pages are the actual containers of information delivered to the user; they are made of *content units*, which are the elementary pieces of information extracted from the data sources by means of queries, and published within pages. In particular, content units denote alternative ways for displaying one or more entity instances. Unit specification requires the definition of a *source* and a *selector*: the source is the name of the entity from which the unit's content is extracted; the selector is a condition, used for retrieving the actual objects of the source entity that contribute to the unit's content. Content units and pages are interconnected by links, which represent navigation alternatives. Links can connect units in a variety of configurations, yielding complex navigation structures. Besides representing user navigations, links between units also specify the transportation of parameters to be used by the destination unit in the selector condition for extracting the data instances to be displayed."

- *Presentation Model* — [Ceri et al., 2000]: "expresses the layout and graphic appearance of pages, independently of the output device and of the rendition language, by means of an abstract XML syntax. Presentation specifications are either page-specific or generic. In the former case they dictate the presentation of a specific page and include explicit references to page content (e. g., they dictate the layout and the graphic appearance of the" attributes of a specific object included in the page; "in the latter, they are based on predefined models independent of the specific content of

[4]http://www.w3.org/XML/

the page and include references to generic content elements (for instance, they dictate the layout and graphic appearance of all attributes of a generic object included in the page).

- *Personalization Model* — describes users and their organization in groups in the form of entities called *user* and *group* and defines personalization based on the data stored in these entities. Their features can be used for storing group specific or individual information. Personalization can be done in two complementary ways [Ceri et al., 2000]:

  (a) *declarative personalization*: "the designer defines derived concepts (e. g., entities, attributes, multi-valued components) whose definition depends on user-specific data" and "the system fills in the information relative to each user when computing the content of units."

  (b) *procedural personalization*: "WebML includes an XML syntax for writing business rules that compute and store user-specific information. A business rule is a triple event-condition-action, which specifies the event to be monitored, the precondition to be checked when the event occurs, and the action to be taken when the condition is found true. Typical tasks performed by business rules are the assignment of users to user groups based on dynamically collected information (e. g., the purchase history, or the access device), the notification of messages to users upon the update of the information base (push technology), the logging of user actions into user-specific data structures (user tracking), and so on."

The main difference between WebML and AHAM is that WebML supports multi-device access to Web applications [Ceri et al., 2000] whereas AHAM deals mainly with the adaptation to the domain model properties. As has been discussed in the AHAM sub-section, adaptation to other than domain model properties (including adaptation to various devices) can be provided in AHAM through adding extra pseudo concepts to the user model. However this is a somewhat artificial solution.

2.5.5 XML Adaptive Hypermedia Model (XAHM)

The XML Adaptive Hypermedia Model (XAHM) [Cannataro et al., 2002] is an XML-based object-oriented model for AHS. XML was chosen as the basic formalism due to its flexibility and data-centric orientation. The (heterogeneous) data sources are modeled in XAHM by means of XML meta-descriptions.

As the basic components the model has the *Application Domain Model* (the hypermedia basic contents and their organization to depict more abstract concepts), the *User Model* and the *Adaptation Model*. The logical structure and contents of an adaptive hypermedia application are described along two different layers. The aim of the lower layer is to define the content of XML pages and the associated semantics (using an object-oriented model based on the class diagrams of UML) and navigational features of the hypermedia (with a directed graph model). The upper layer describes the structure of the hypermedia as a set of views associated to groups of users (i. e., stereotype profiles) and some semantic relationships among profiles (using logical rules). Finally, the adaptation model is based on a multidimensional approach — each part of the hypermedia is described along three different adaptivity dimensions, each related to a different aspect of user's characteristics:

- user's behavior (browsing activity, preferences, etc.);
- external environment (time-spatial location, language, socio-political issues, etc.);
- technology (user's terminal, client/server processing power, kind of network, etc.).

A view over the application domain corresponds to each possible position of the user in the adaptation space. The XML pages are independent from such position, and the final pages (e. g., HTML[5], WML, etc.) to be delivered are obtained through a transformation that is carried out in two distinct phases, the first one driven by the user's profile and environmental conditions, and the second one driven by technological aspects.

The main emphasis in XAHM is on distinguishing between adaptation driven by user needs and adaptation driven by technological constraints. AHAM puts less emphasis on technological aspects though adaptation to them can be provided in AHAM through adding extra pseudo concepts. This is however, as explained earlier, an artificial solution. Another difference between AHAM and XAHM is the usage of overlay and stereotype user model (see 2.3), respectively.

2.5.6 Summary of models for adaptive hypermedia

The review of adaptive hypermedia models shows a number of differences between them. The major differences are:

- a clear separation between the adaptation to domain model related aspects, such as user's knowledge of the domain model concepts, and technological aspects, such as device, kind of network, that exist in WebML and XAHM but does not exist in AHAM;
- an extra goal and constraints model introduced by LAOS.

The first issue can be addressed in AHAM through adding extra pseudo concepts reflecting the technological aspects, though this is considered to be an artificial solution. Furthermore for the research described in this dissertation this type of adaptation is not crucial as we will be dealing with adaptation to learning styles.

LAOS extends the AHAM model with an extra goal and constraints layer to separate purely domain-dependent information from additional pedagogic information added by the teacher. This is an important extension, however, it also has a side effect that an author has to design an extra layer.

While the above mentioned models provide additional features that AHAM does not have, for the reasons we give, we consider that following AHAM is sufficient for our research.

[5]http://www.w3.org/html/

2.6 Some Examples of AHS

In this sub-section we give an overview of a number of most influential adaptive hypermedia systems. Presenting more systems is out of scope of this dissertation. We have chosen to present InterBook [Brusilovsky et al., 1998; Eklund and Brusilovsky, 1999] as it is de facto standard in the field of adaptive hypermedia. We also present KBS Hyperbook [Nejdl and Wolpers, 1999; Henze and Nejdl, 1999, 1998] as it provides a very different approach to modeling the user than most of the AHS do. Both systems started approximately in the same period as AHA!. Later on we present APeLS [Dagger et al., 2003b,a; Canavan, 2004] which is the more recent AHS.

2.6.1 InterBook and the history leading up to InterBook

We first have a look at the evolution of several systems leading up to InterBook (see figure 2.3 taken from the slides of the invited talk given by P. Brusilovsky at the 5th International Conference on Intelligent Tutoring Systems, Montreal, Canada, 2000, http://aied.inf.ed.ac.uk/members01/archive/vol_12/brusilovsky/sld002.htm).

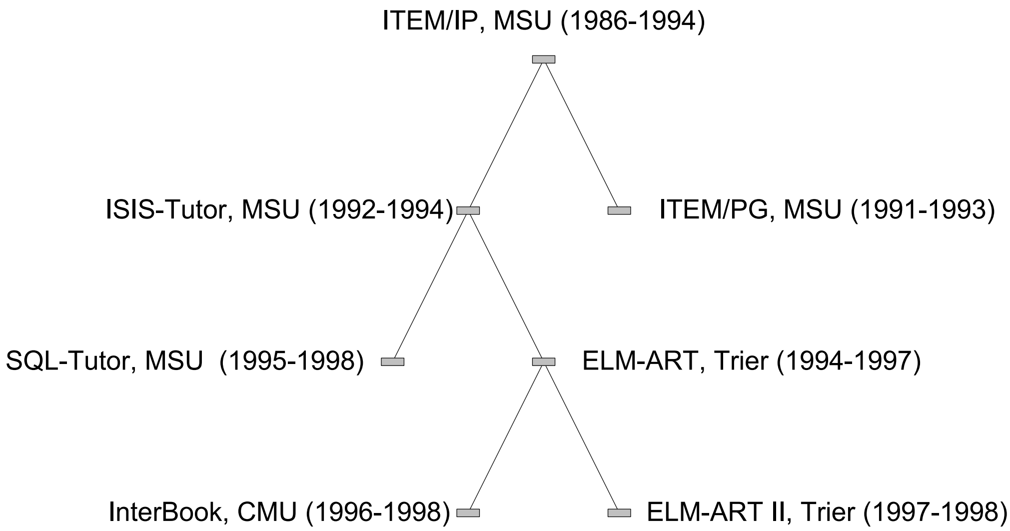


Figure 2.3: History of AHS development leading up to InterBook

The evolution started from **ITEM/IP** (Intelligent Tutor, Environment and Manual for Introductory Programming) developed at the Moscow State University (MSU) in Russia [Brusilovsky, 1992]. ITEM/IP integrates an intelligent tutoring system for programming, a programming environment, and adaptive on-line help facilities on the basis of the domain model and the overlay student model which serve as a kernel of the system. The domain model is a network of concepts. Nodes and links of the network represent elements of domain knowledge and relations between the elements. The student model represents the level of the student's competence for each of the knowledge elements [Brusilovsky and Pesin, 1994].

The descendent of ITEM/IP, **ISIS-Tutor** [Brusilovsky and Pesin, 1994], is an intelligent hypertext learning environment intended for learning the print formatting language of an information retrieval system CDS/ISIS/M (briefly - ISIS, supplied by UNESCO) for IBM PC. The ISIS-Tutor was developed at the International Centre for Scientific and Technical Information (ICSTI) and Moscow State University. The system has an interrelated domain model and overlay student model. In difference with ITEM/IP the teaching material in ISIS-tutor forms a real hypermedia network. The guided tutoring mode (provided by the tutor) and free exploration mode (provided by the hypermedia) are integrated in ISIS-tutor. The tutor component supports adaptive task sequencing, which means that knowledge demonstrated by the student in the past is analyzed and the system selects an optimal teaching operation (such as concept presentations, examples and problems) to perform. The hypertext component of the system supports student-driven acquisition of conceptual knowledge. It is an integrated part of the system. It means that the component uses the student model to provide adaptive navigation support for the given student, and it updates the student model to reflect the results of the student's work with the component.

The evolution continued with **SQL-Tutor** [Mitrovic, 1997] developed at Moscow State University, and with **ELM-ART** [Brusilovsky et al., 1996] developed at the University of Trier in Germany. ISIS-Tutor was developed in Pascal while both SQL-Tutor and ELM-ART were developed in Lisp. InterBook [Brusilovsky et al., 1998], a descendent of ELM-ART, was developed at the Carnegie Mellon University (CMU) in Pittsburgh, USA. It is also implemented in Lisp.

ELM-ART (ELM Adaptive Remote Tutor) [Brusilovsky et al., 1996] is a Web-based intelligent tutoring system to support learning programming in Lisp. ELM-ART can be considered as an on-line intelligent textbook with an integrated problem solving environment (the authors call it I^3 textbook, or **I**ntelligent **I**nteractive **I**ntegrated textbook). It provides all the course materials (presentations of new concepts, tests, examples, and problems) in hypermedia form. To represent the conceptual knowledge of individual students the system uses a traditional overlay student model.

To support the student navigating through the course, the system uses adaptive annotation and adaptive sorting of links. The system also provides prerequisite-based help in two cases. First, when a student enters a page which is not yet ready to be learned, the system warns the student that this material has unlearned prerequisites and shows additional links to textbook and manual pages where the unlearned prerequisite concepts are presented. Second, when the student has problems with understanding some explanation or example or solving a problem, (s)he can request help and, as an answer to help request, the system will show the links to all pages where the prerequisite knowledge is presented.

Another important feature of ELM-ART is that the system can predict the student's way of solving a particular problem and find the most relevant example from the individual learning history. Answering the help request ELM-ART selects the most helpful examples, sorts them according to their relevance (from most to least relevant), and presents them to the student as an ordered list of hypertext links.

If the student failed to complete the solution of the problem, or if the student cannot find an error which was reported by the testing component, (s)he can ask the system to diagnose the code of the solution in its current state. As an answer, the system provides a sequence of help messages with more and more detailed explanation of an error. The sequence starts with a very vague hint at what is wrong and ends with a code-level suggestion on how to correct the error or how to complete the solution. In many cases, the student can understand where the error is or what can be the next step from the very first messages and will not need more explanations. The solution can be corrected or completed, checked again, and so forth. The student can use this kind of help as many times as required to solve the problem correctly. In this context, the possibility to provide the code-level suggestion is a very important feature of ELM-ART as a distance learning system. It ensures that all students will finally solve the problem without the help of a human teacher.

InterBook [Brusilovsky et al., 1998; Eklund and Brusilovsky, 1999] is a tool for authoring and delivering *adaptive electronic textbooks* on the World Wide Web.

[Eklund and Brusilovsky, 1999]: "Any knowledge base that contains reasonably specific and identifiable knowledge elements that can be organized hierarchically into sections, subsections and indexed in detail is suitable for delivery through the InterBook system. Technical and software manuals are an excellent example of suitable material. InterBook takes the structures commonly found in such a textbook (such as tables of content, indexes and glossaries) and delivers them on the web with navigation support, providing individualized assistance to each learner."

The domain model in InterBook consists of concepts to be learned and prerequisite relationships between these concepts. [Eklund and Brusilovsky, 1999]: "All student actions (page visits, problem-solving, quizzes answering) are tracked and used to increase or decrease knowledge levels for involved concepts. For each domain model concept, an individual student's knowledge model (overlay model) stores some value which is an estimation of the student knowledge level of this concept. InterBook distinguishes different knowledge levels: unknown, known, learned and well-learned. Another component of the student model is the model of a student's individually assigned learning goal. Adaptive guidance mechanisms will ensure that the student achieves a sequence of assigned learning goals."

[Eklund and Brusilovsky, 1999]: "All InterBook-served electronic textbooks have a generated table of content, a glossary, and a search interface. In InterBook, the structure of the glossary resembles the pedagogic structure of the domain knowledge in that each node of the domain network is represented by a glossary entry. Likewise each glossary entry corresponds to one of the domain concepts. All sections of an electronic textbook are indexed with domain-model concepts. InterBook provides an adaptive set of links between the text and the glossary based on the current user's knowledge."

[Eklund and Brusilovsky, 1999]: "InterBook uses colored bullets and different fonts to provide adaptive navigation support through link annotation. Annotations are evident in the individual section of the text (in the textbook window) as well as in a local overview map (on the navigation bar). Wherever a link appears on a page (in the table of content, in the glossary or on a regular page), the font and the color of the bullet informs the user about the status of the node behind that link. Four colors and three fonts are used. A green bullet and bold font means "ready and recommended", i. e., the node is ready-to-be-learned but still not learned and contains some new material. This also means that all the prerequisite concepts for that node have been visited. A red bullet and an italic font warn about a not-ready-to-be-learned node meaning that it contains prerequisite concepts that have not been visited. White bullet means "clear, nothing new", i. e., all concepts presented on a node are known to the user. Violet is used to mark nodes which have not been annotated by an author. A check mark is added for already visited nodes."

Besides links annotation InterBook provides direct guidance (through "Teach me" button) about the suggested next place the student should visit. Another kind of direct guidance is used to provide prerequisite-based help for the user. Since the system knows the prerequisite relationships between concepts, when the user has difficulty understanding a concept or solving a problem, the system can suggest the unit that contains the concepts that are the prerequisite concepts of the difficult unit.

2.6.2 KBS Hyperbook

The KBS Hyperbook system [Nejdl and Wolpers, 1999; Henze and Nejdl, 1999, 1998] was developed at the Institut für Rechnergestützte Wissensverarbeitung in Hannover, Germany, whose English name "Knowledge Based Systems" (KBS), gave the name for the project. KBS Hyperbook is a tool for modeling, organizing and maintaining adaptive, open hypermedia systems on the World Wide Web. Open in this context means that these hypermedia systems are able to integrate distributed information. The implemented hyperbook is used for a course "Introduction to Programming in Java".

Part of the information units used in the KBS Hyperbook resides on the developers' server another part is being integrated from the online SUN Java Tutorial and the Java API documentation [Henze and Nejdl, 1999].

The authors emphasize the importance of teaching strategies that enable a learner to actively learn and not only to passively read or "consume" the information. They follow a constructivist pedagogic approach that involves project-based learning, group work and discussions. To support this type of learning KBS Hyperbook implements the following adaptation components [Henze and Nejdl, 1998] (comprising the *adaptation model*):

- *adaptive information resources* — give the students appropriate information while performing their projects, by annotating necessary project resources depending on current student knowledge;

- *adaptive navigational structure* — adapt/annotate the navigational structure in order to give the student additional information about appropriate material to explore/learn next;

- *adaptive trail generation* — provide guidance by generating a sequential trail through part of the hyperbook depending on student goals;

- *adaptive project selection* — provide suitable projects depending on student goals and previous knowledge;
- *adaptive goal selection* — propose suitable learning goals depending on users knowledge.

The main content of the hyperbook (the *domain model*) consists of semantic information units and project units. [Henze and Nejdl, 1998]: "Both of these contain actual content as a WWW page or as a sequence of WWW pages". "Information units do not correspond to syntactical parts of a book (such as sections or chapters), but semantical parts (such as information units about Java Objects, Iteration Constructs, Parameters, etc.). Information units are indexed by knowledge items." Such a knowledge item (**KI**) denotes an elementary knowledge concept, the set of items describes the knowledge of the application domain. In many cases there is a one to one correspondence between information units and knowledge items. "One or more of the knowledge items belonging to a page are the main knowledge items of this page, and for each knowledge item there is exactly one information unit, where it is a main knowledge item. This leads to a kind of knowledge item index, which gives for each knowledge item one main information unit, and some other information units where it occurs too, but not as main knowledge item."

As information units are semantic entities, they are semantically related to other information units, for example, through "is-a" (specialization) or "instance-of" (instantiation) relationships. These semantic relationships generate the navigational structure between the information units (which is done dynamically by the KBS Hyperbook system), so each link between information units corresponds to some kind of semantic relationship between these units. This navigational structure can be annotated (already known, suggested, too difficult) according to the current knowledge of the reader (adaptation component *adaptive navigational structure*).

A simple traffic light metaphor is used for annotation [Henze and Nejdl, 1998]: "a red ball in front of the link indicates that the corresponding page requires some knowledge the user currently does not have and thus is not recommended for the user (too difficult), while a green ball denotes a recommended link (suggested), which should be understandable for the user. Finally, a grey ball (already known) denotes material which (according to the hyperbook's estimate of the user) is already known to the user.

Project units represent project descriptions, and are indexed by those knowledge items which the student needs to know in order to successfully work on these projects. The relationship between project units and information units can be automatically derived (via the knowledge items) and shows the information units which are relevant for a given project. The links corresponding to this relationship can be adapted as well. This is done by annotating the links according to the users knowledge (already known, suggested, too difficult), leading to an *adaptive information resource* for a given project. The annotated links are shown as an annotated index (from the project unit to the corresponding information units). The system can also generate a sequential trail (guided tour) through these information units, leaving out already known information units, and ordering the remaining information units, such that difficult information units are suggested at a later stage, when the user knows enough in order to understand them (*adaptive trail generation*)."

Furthermore, [Henze and Nejdl, 1998] "the user can select a set of knowledge items (called a goal), and the system can generate (according to the users knowledge) an index of projects most useful for achieving the users learning goal (*adaptive project selection*), a trail for learning these knowledge items (adapted to the users knowledge) or an annotated index of information units for this goal. Finally, the hyperbook system can propose suitable learning goals for the user based on the users current knowledge (*adaptive goal selection*), and then propose corresponding projects, trails or information units."

[Henze and Nejdl, 1998]: "The *user model* underlying the hyperbook is a kind of overlay model using a Bayesian Network (BN) for estimating user knowledge." It contains the knowledge items **KIs** used in the domain model of hyperbook and adds a partial order between them to represent learning dependencies. For example, **KI1** < **KI2** denotes the fact that **KI1** has to be learned before **KI2** because understanding **KI1** is a prerequisite for understanding **KI2**. The user model also contains descriptions of each users current knowledge in the form of a knowledge vector. This decoupling between hyperbook model and user model has advantages for authoring the hyperbook, as learning dependencies between knowledge items are described once in the user model, and the dependencies between information units of the hyperbook can be inferred automatically from the **KI**-dependencies and the indexing of the information units by the **KIs**. Furthermore the user model includes an inferring mechanism for calculating the system's beliefs about a user's knowledge of these items (Bayesian Network) [Henze and Nejdl, 1999].

[Henze and Nejdl, 1998]: "BNs are useful in user modeling, since they allow to describe the application domain in a single dependency graph. This graph contains all necessary prerequisites for a particular knowledge item, models dependencies among knowledge items and is able to infer, for example, that prerequisite knowledge of a **KI** has already been acquired by a user if the **KI** itself is understood by the user."

Another advantage of using BNs is the handling of uncertainty in observations. It is possible to: "use every degree of information about the users knowledge, not only failed/not failed." The system uses a vector of four probability values (summing up to 1) describing the estimate that a user understands a specific knowledge item to the degrees excellent (expert user), some difficulties (advanced user), many difficulties (beginner), not ready (newcomer).

The use of Bayesian Network technology in KBS Hyperbook is distinct from other approaches to student and user modeling [Henze and Nejdl, 1998]. InterBook and AHA!, for example, use the vector representation of the overlay user model. In InterBook, the student model is a vector that stores the status value for each concept page and each book page [Brusilovsky and Sampson, 2004]. In AHA! this is a vector with the attribute-value pairs for all domain model and pseudo concepts.

KBS Hyperbook is different from many other AHS in a few more aspects.

In most other adaptive hypermedia systems, dependencies like prerequisites or outcomes are directly connected to the information resources themselves [Henze and Nejdl, 1998]. In ELM-ART and InterBook information about the reading sequence is represented through the domain model concepts hierarchy, while as has been shown earlier, KBS Hyperbook stores dependency information in the user model. Furthermore, the information units in KBS Hyperbook are not only related in a hierarchical way, but can be arbitrary relationships. We will see in the following chapter that AHA! allows for arbitrary concept relationships as well.

KBS Hyperbook does not take into account information about the visited pages or users' paths through hypertext (like, for example, InterBook and AHA! do). Instead the system directly asks the user for feedback on different topics (**KI**s) after each project unit. The user can choose between different answers such as "topic was easy - I mastered it effortlessly", "topic was okay - but I had some problems, "topic was hard - I had a few ideas but could not get the solution and "no idea about this topic at all" [Henze and Nejdl, 1998].

Finally, KBS Hyperbook implements a number of adaptation tasks that are based on active, project-based learning as opposed to passive ones [Henze and Nejdl, 1998].

2.6.3 APeLS (Adaptive Personalized e-Service)

APeLS [Dagger et al., 2003a,b; Canavan, 2004] was developed by the Knowledge and Data Engineering Group in the Department of Computer Science at Trinity College Dublin, Ireland. The three main models in APeLS are the *learner*, *content* and *narrative* (pedagogical) models. These models correspond to user, domain and adaptation models of the AHAM model. The adaptive engine of APeLS is being fed by these three models [Dagger et al., 2003b].

For *modeling learner*, APeLS primarily relies on direct feedback technique to capture information for updating the learner model (e. g., submitting questionnaire answers) rather than indirect feedback technique (such as the results of exercises or problem solving tasks or the learner's browsing behavior). The student is categorized by stereotype and the model is gradually modified as the student moves through the course [Canavan, 2004].

For *modeling content*, APeLS implements the concept of a candidate content group (CCG). The function of a CCG is to group together learning resources that are equivalent on some axis, for example, concept taught, prerequisites or learning style. Each CCG has metadata associated with it to describe the role of the group and to identify the learning resources associated with it [Canavan, 2004]. The learning resources themselves are known as learning objects (LOs)[6]. This separation of content from narrative facilitates the reuse of the LO. Each LO is made up of two elements: learning content (e. g., piece of text or picture) and metadata to describe it. The descriptive metadata provides a means of searching for LOs in the content repository and also provides information to the adaptive engine when content needs to be selected for the presentation to the learner.

The primary goal of the *narrative* is to produce personalized courses that are structured coherently and which cover the learner's goals. The narrative captures the logic behind the selection and delivery of a learning resource within the scope of the adaptive

[6]http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

course [Dagger et al., 2003a]. It allows the author to separate the intelligence that performs the adaptation from the content and this increases the potential reuse of the learning resources involved [Dagger et al., 2003a]. The narrative therefore refers to the CCG rather than to the learning resource directly thus achieving a level of abstraction. In APeLS it is possible to create a narrative that encapsulates different pedagogical approaches through writing several candidate narratives which achieve the same learning objectives. Appropriate selection mechanisms are added to narrative model. The course is formed (or compiled) whenever the user requests this action to occur. The action can be invoked at any stage by the learner.

APeLS provides adaptive presentation through candidacy. Candidate selectors can be defined as the rule sets that choose a candidate from a CCG [Dagger et al., 2003a]. Candidate selectors can be implemented in two modes — all (set of candidates that meet the minimum requirements) and best (single candidate that best fits the requirements). When there are several selectors each one called refines the list of candidates with the last one called selecting the best candidate from the final list. APeLS facilitates adaptive navigation through the sequencing of candidates in the narrative. It is possible to have several candidate narratives for a single course, for example, a course may be structured so that the candidates have the same ethos, learning goals and require the same prior knowledge but differ in pedagogical approach [Conlan et al., 2002]. It would be possible, for example, to develop two narratives for a course, one which is case-based and another didactic. This methodology would support adaptive navigation by sequencing the material differently for each of the candidate narratives. At runtime the candidate narrative most suited to the learner would be selected [Canavan, 2004].

2.6.4 Summary of AHS examples

Adaptive hypermedia research provides a large number of special-purpose AHS, however, up to now we did not manage to find a tool targeting various application areas. In this section we presented a number of adaptive *educational* hypermedia systems. Nevertheless these AEHS give quite a broad overview of the adaptive hypermedia field since they are very different in several aspects (we only name here the major differences):

- methods for modeling the user — overlay in InterBook and KBS HyperBook and stereotype in APeLS;
- technologies for modeling the user — vector representation in InterBook and Bayesian networks in KBS Hyperbook;
- representation of the domain model — concepts hierarchy in InterBook, semantic information units and project units in KBS Hyperbook, candidate content group in APeLS;
- defining adaptation — in InterBook and KBS Hyperbook adaptation is pre-defined, in APeLS an author can write own narratives encapsulating different pedagogical approaches.

Furthermore in section 5.2 of chapter 5 we present the examples of AHS providing adaptation to learning styles. In chapters 3 and 6 we present how the AHA! system fits the adaptive hypermedia field.

2.7 Authoring Adaptive Hypermedia

Initially AHS were mostly focused on the delivery of an adaptive application to the end-user and less on the authoring aspects. However to allow widespread use of adaptive hypermedia systems the difficulty of the authoring process should be considered and ways to make the authoring process as "simple" and intuitive as possible should be found.

Typically authoring of an adaptive application involves [De Bra et al., 2003b; Stash et al., 2005]:

- *creating application content* — including creating alternatives of the same content and structuring application content in a way that makes multiple navigation paths possible to cater for different users' needs;
- *creating domain model* — creating the structure (including the attributes) and the hierarchy of the domain model concepts connected to the application content pages;
- *specifying aspects used for adaptation* — such as user data, usage data or environment data (see section 2.3);
- *creating adaptation model* — defining how the user model should be updated and how the adaptation should be performed.

Thus authoring of adaptive hypermedia is often considered to be a difficult, time-consuming and therefore also expensive process [Stewart et al., 2005]. Whereas this is certainly true when one looks at the complete authoring process, most of the effort actually goes into the creation of content (or the information pages). In this dissertation we do not consider this part of the authoring task but concentrate on the aspect of building the adaptive functionality.

We identify the following steps that should be taken in order to support the authoring process:

1. *Authoring tools* that make authoring of adaptive applications easier — authors (especially those who are not computer experts) need a user friendly (hence probably graphical) authoring environment for designing the adaptation aspects of their applications.
2. *Reuse of content* — the content of the domain model may consist of information items that were created before, for other environments. The authoring tool for the adaptive application must not require this material to be re-entered, but must allow importing it from other sources.

3. *Avoiding repetitive work* — the adaptation is often based on constructs that appear many times (in identical form). An example is the adaptation based on prerequisite relationships. Defining this adaptive behavior should be done once and repeating it for all instances should be possible with minimal effort.

Different systems take a different approach towards authoring. Each AHS system uses its own content model, adaptation coding methodology and presentation style. Here we present two approaches — an InterBook approach due to its simplicity and an authoring environment MOT which supports the definition of a more complex adaptive behavior.

2.7.1 Authoring adaptive electronic textbooks in InterBook

For creating an electronic textbook, InterBook [Brusilovsky et al., 1998] requires authors to prepare a specially structured Microsoft Word file which will be later converted into InterBook format. The resulting textbook file can be served on WWW by the InterBook system. Authoring an adaptive electronic textbook consists of 5 steps described below.

The first step requires creating a properly structured Microsoft Word file. InterBook recognizes the hierarchical structure of the document through the use of headers. This means that the author should use a predefined style for the titles of the sections at different levels, e. g., "Header 1", "Header 2" and so forth.

The second step involves adding annotations to the previously created file to let InterBook know which concepts stand behind each section. Each section is annotated with a set of prerequisite concepts (concepts which should be read before the current section) and a set of outcome concepts (concepts which are supposed to be known after the user has visited the section).

Once the annotations are complete the third step is to save the file in the RTF format.

Fourth, a special RTFtoHTML program is used for converting RTF to HTML format. Then ".html" extension of the file is manually altered to ".inter" so that it can be recognized by the InterBook system.

Finally, the InterBook server parses this file and creates the textbook pages. The content which is presented to the student is generated on-the-fly using the knowledge about the textbook, the student model, and the created textbook pages.

The use of Microsoft Word makes authoring easier as the author only needs to know Word and the special annotations. An advanced author who has some knowledge of HTML and LISP programming can use the tool more flexibly. For example, (s)he can skip the first two steps of authoring a textbook in Microsoft Word and prepare the textbook directly in HTML format with annotations provided as specially formatted comments. Furthermore the author can overwrite server response functions and HTML generating functions to implement different structure and different "look and feel" of the generated pages.

For InterBook an author thus creates only the domain model and application contents. Despite its easiness the authoring approach in InterBook has one disadvantage: the (conceptual) domain model and application contents (the pages) are mixed.

2.7.2 Authoring adaptive applications in MOT

MOT (My Online Teacher) [Cristea and de Mooij, 2003] aims at establishing a "write once, use many" methodology for the creation of content and adaptation within current adaptive educational hypermedia (AEH) environments [Stewart et al., 2004]. MOT is an online (client-server) environment for the authoring of adaptive educational hypermedia. It is based on the theoretical frameworks LAOS and LAG presented in 2.5.3. MOT provides an interface for implementing the following layers:

- *Domain Maps.* At this step the author defines the concept hierarchy and the attributes of the concepts. It is possible to define non-hierarchical types of relationships. MOT provides a default set of concepts' attributes such as title; keywords; introduction; pattern; text; explanation; conclusion and exercise. Filling in the title attribute is compulsory while the other attributes are optional. Additional attributes can be added by the author if desired.

- *Goal and Constraints Maps.* In MOT, goals and constraints maps (also called lesson maps) are implemented as a simplified overlay over the domain maps. Only in these lesson maps the concepts are ordered depending on pedagogic constraints and teaching strategies. This lesson contains selections filtered from the domain model based upon instructional views and goals (such as expected time frame, level, background knowledge, learning styles, etc.). The author can select either all or only certain attributes of the domain model concepts for the inclusion into the lesson. Furthermore the lesson stores the information about the relevance of a certain resource to the different types of learners (e. g., learners with textual, mixed or visual preferences). For this the author has to specify the labels for concepts' attributes showing which attributes (resources) are to be delivered to a certain type of learner and weights which can be used for a more refined tuning of the classification.

- *User Maps.* In this step the author defines all the necessary variables and their initial values that are necessary to represent the user. User model variables are defined as:

  - overlay user map variables declaration, e. g., the knowledge variable for every concept in the lesson map;
  - independent user map variables declaration, e. g., the stereotype variable for any user.

  The initial values for all variables can be, e. g., 0.

- *Presentation Maps.* At this step the author defines all the necessary variables and their initial values that are necessary for the presentation of the lesson to the learner. Presentation model variables are defined as:

  - overlay presentation map variables declaration — e. g., the display ("show") variable for every concept in the lesson map;

  – independent presentation map variables declaration — e. g., the Boolean variables determining if a table of contents ("Menu"), "Next" button or "To Do" list items appear in the presentation.

  For the concepts without specific semantic labels for their attributes, the initial values for the showability variables ("show") can be set to "true" so that they can be visualized by any type of user. Also the table of contents and "To Do" list variables can be set to "false", resulting in a linear presentation for the learner (when the learner only needs to press the "next" button to proceed through the course).

- *Adaptation maps.* In this step the author defines how the values of the variables in the user and presentation maps are changed during the interaction of the learner with the system. The Adaptive Strategy Interface [Stewart et al., 2004] in MOT is built based on the three layers of adaptation granulation theoretical framework LAG. It allows the author to use the LAG language to define adaptive strategies that can represent teaching or pedagogic strategies. The adaptation strategies build the actual adaptation maps and can be reused in different contexts and for different combinations of domain and lesson maps.

2.7.3 Summary of authoring adaptive hypermedia

In this section we presented two approaches to authoring adaptive hypermedia. We have seen that authoring can be quite easy as in the case of using the InterBook system. In InterBook, the author does not have to define the adaptation model. The adaptation is pre-defined in the system. On the other hand, MOT adds more complexity but provides more flexibility by allowing authors to design their own pedagogic strategies comprising the adaptation model. As will be discussed in section 6.1 of chapter 6, the adaptation language used for defining the strategies provides the increased level of semantics therefore allowing for more than IF-THEN rules in the strategies. MOT separation of layers being authored allows for reuse of information. For example, the strategies designed by a certain author can be reused by other authors in different applications.

MOT currently does not provide a delivery engine. A transformation from MOT to other systems has been described in the following papers: MOT to AHA! conversion in [Cristea et al., 2005a], MOT to WHURLE conversion in [Cristea et al., 2005b].

In section 3.3 of chapter 3 and section 6.2 of chapter 6 we describe how authoring is done in the AHA! system.

2.8 Design Requirements for a General-Purpose System

In the previous sections of this chapter we provided background information on the adaptive hypermedia field. This information helps us to derive the design requirements for a general-purpose system which we formulate here.

Most existing AHS are *special-purpose* tools, meaning that they are geared towards one specific application or application area, in most cases related to education. In this dissertation we ask ourselves the question whether an AHS is necessarily a special-purpose tool? Can we create a system that would allow creation and delivery of various types of adaptive applications targeting different application areas? We need to make the difference between the terms *general-purpose* and *all-purpose* tool clear. We cannot provide an all-purpose tool which everyone could use for every desirable purpose. We cannot foresee all possible expectations of the authors for the adaptive behavior of their applications, different types of strategies and concept relationships they might want to apply. We are therefore talking about a general-purpose tool which can serve many different (but not all possible) purposes.

We start answering the first research question by specifying the features the desired tool should satisfy. As discussed in 2.5.6, we want this tool to be as close as possible to the AHAM model. By analyzing AHAM presented in section 2.5.1 and looking back at sections 2.2 ("Application Areas of AHS"), 2.3 ("Modeling the User") and 2.4 ("Methods and Techniques Used in AHS") of the current chapter we formulate the design requirements as follows:

Design requirement "adaptation types and techniques"

This design requirement has been derived from section 2.4. A general-purpose system should be able to provide the following types of adaptation.

- content adaptation,
- link adaptation,
- layout adaptation.

Furthermore, the system should allow the implementation of adaptive hypermedia techniques as presented in figure 2.1. Most often the existing systems do not provide all three types of adaptation and they implement only few techniques — mostly link annotation or hiding and/or canned text adaptation as described in Brusilovsky terminology [Brusilovsky, 2001]. Adaptive layout is very rare except in research prototypes that focus on adaptation to desktop and mobile devices, as, e. g., in prototype multimedia document transformation environment Cuypers [van Ossenbruggen et al., 2001]. In more common AHS, you are lucky if you have adaptable layout in some form (like in AHA! in which layout can be adapted, but only by the author and not by the end-user).

Design requirement "aspects used for adaptation"

This design requirement has been derived from section 2.3 and from the description of the AHAM domain model. A general-purpose system should be able to perform adaptation based on three types of information:

- the relation between the user and the concepts in the domain model. It should be possible to perform adaptation based on multiple attributes of the concepts, such as the knowledge about or interest in a concept, the number of visits to the concept, etc..

- arbitrary user's characteristics that are not about the domain model, for example, background experience, or the user's learning style. The user's characteristics may lead to adaptation of the information, its presentation and the recommended order in which to visit the concepts.

- aspects of the user's environment, including place, time, used equipment (computer, PDA, TV) and network. The adaptation may involve changes in media (like eliminating sound in an open office), in quality and/or resolution (for small screens and/or slow networks) but possibly also to the information itself (e. g., eliminating some unsuitable content in an environment where children may be present).

We discuss further in more detail the first aspect concerning the user's relation to the domain model when browsing the hyperspace.

Typically, AHS require the presence of specific attributes in the user model to perform adaptation [Wu, 2002]. Many AHS store at least the following two attributes, and maybe also the third one [De Bra et al., 1999]:

- *knowledge* attribute — indicates how much knowledge the user has about the concept. Educational AHS are based on the notion that the user model consists of concepts with an associated knowledge value. Each time a page (about a concept) is accessed some server-side program (e. g., a Java Servlet) registers this access and updates the knowledge value for the associated concept.

- *read* or *visited* attribute — indicates whether the user has read something (a fragment, a page or a set of pages) about the concept. In Web-based systems, the *read* attribute is used to generate a different presentation for anchors of links to pages that have been read than for links to unread pages. (By default, in most browsers the difference is a purple versus blue color for the anchor text or image border). The name *read* suggests a Boolean value but the AHS may also use a counter to keep track of the number of visits to a concept.

- a less common attribute *ready* — sometimes also referred to as *desirable*, *recommended* or *suitability*. It indicates whether the user is ready to read about the concept this attribute is bound to. For example, in a learning application this would mean that sufficient prerequisite knowledge has been acquired.

A general-purpose system should not require the presence of specific attributes in the user model. Therefore to turn an AHS into a general-purpose one the notions of *knowledge about a concept*, *read* or *ready status*, etc. must be generalized to *value for a user model attribute*. As such, a concept need not represent domain knowledge, and the attribute may represent any kind of property. A general-purpose AHS should be able to perform adaptation based on the user's browsing actions, regardless of interpretation of browsing as a particular type of activity, e. g., learning in educational AHS [De Bra et al., 2000].

While most AHS provide a fixed set of attributes, a general-purpose AHS should allow authors to "invent" new attributes which they might need for their applications.

The attribute values in most of the systems are of simple types — integer, boolean and string. So complex, and possibly even recursive structures, are typically not possible. In order to develop a truly versatile system it should be augmented with facilities to define a user model with arbitrarily many attributes of arbitrary data types for each concept.

Design requirement "arbitrary concept relationships"

This design requirement has been derived from section 2.2 and from the description of the AHAM domain model. A general-purpose system should allow arbitrarily many types of concept relationships.

In actual AHS there is usually a fixed set of concept relationships types. They may limit the use of AHS to a certain kind of application. For authors who only need the existing types of relationships the creation of an application is easy. But when a different type of relationship is needed authors may have to play "tricks" with the available types to create the adaptation they want. In educational settings, the most commonly used types of relationships are *knowledge propagation* and *prerequisites*. In a museum there may be *exemplifies* relationships between paintings and a *painting style* or a *painted by* relationship between paintings and painters. In an on-line shop products may be *competing* with other products or may be *complementing* other products.

A general-purpose tool should allow various types of concept relationships in order for it to be used to realize various types of applications.

Design requirement "generic adaptation rules"

This design requirement has been derived from the description of the AHAM adaptation model in section 2.5.1. A general-purpose rule system should allow a "generic" way of defining adaptation rules.

The adaptation that an AHS can perform may be partly built-in, defined by a system designer, and partly defined by an author. In order to make authoring as easy as possible all application-independent adaptation rules should be defined by a system designer, leaving out some application dependent rules to the author. As has been already discussed in the AHAM section, AHAM makes a distinction between *generic* and *specific adaptation rules*. System-defined rules will be almost always generic adaptation rules. Authors may wish to write some specific rules to create uncommon adaptation linked to a few specific concepts, thereby possibly overriding the adaptation defined by some generic rules. Still, authors should be also able to define generic rules that they can apply to all concepts or all concept relationships of a certain type. However it becomes difficult to predict the behavior of the system when author-defined rules are present. The rule execution (within a phase) may not terminate, or if it does it may not always produce the same (predictable) result. These problems are generally called *termination* and *confluence* [Wu, 2002]. They are addressed in the following design requirement.

Design requirement "handling non-monotonic user model updates and cycles in adaptation rules"

(This design requirement has been derived from the paper about AHAM "Design Issues for General-Purpose Adaptive Hypermedia Systems" [Wu et al., 2001].) A general-purpose system should be able to handle *non-monotonic* user model updates and *cycles* in adaptation rules.

Typically in an educational system a user's knowledge about a concept can only increase. Possible exception is when a learner performs badly on a test about this concept. However during learning, knowledge increase (as a consequence of remembering) is not the only process [Ágh and Bieliková, 2004]. The learner can also lose (e. g., forget) some knowledge in the course of the time since the acquired knowledge is not stored in the human memory forever. Knowledge forgetting can cause inconsistencies between the system's assumptions about the user (represented through the user model) and the actual state of the user's knowledge [Ágh and Bieliková, 2004]. Thus it is important to provide a way for decreasing knowledge in an educational system as well. One approach to this is described in [Ágh and Bieliková, 2004] where the authors propose a model which reflects the forgetting.

Another example that requires non-monotonic updates — inferring user's preferences for different types of information available in an application. Access to a certain type of information (e. g., pictorial representations of the concepts) when different types are available can be considered as an increase of interest in this type and at the same time as a decrease of interest in other types of information (e. g., textual representations of the same concepts). Thus increase/decrease of interest in each separate type of information should be possible.

Therefore one of the design requirements for a general-purpose tool is the ability to handle *non-monotonic* user model updates.

Adaptation features depend heavily on the updates to the user model being performed correctly. Badly designed user model update algorithms can produce non-deterministic results or infinite loops. Therefore another requirement for a general-purpose tool is the ability to handle *cycles* in the adaptation rules.

In [Wu et al., 2001] the authors concentrate on the architecture and behavior of a *general-purpose adaptive engine* of the AHAM model and study the above mentioned issues of termination and confluence. The AHAM suggestion is to use activation graphs (from the static analysis of active database rules), thereby constructing the whole graph of possible states that is determined by the concepts, their links, the attributes, their values (especially, initial values and possible ranges to eliminate unnecessary branches and optimize the search tree with the help of constraints) and the rule sets. If such a graph has no cycles, the system will always terminate. For confluence, a difficult procedure of checking the possibility of commutation between each rule pair (so their order equivalence) is proposed.

Other issues/requirements

One of the design goals for a general-purpose system is reasonable performance. When the execution of the adaptive rules takes a long time, there will be a very noticeable delay between the "click" on a link anchor and the presentation of the link destination page. The adaptation engine needs to deal with the problems of termination and confluence. However the system designers also need to ensure that, whichever measures are taken to guarantee termination and confluence, rule execution never takes a long time while the user is browsing.

Based on section 2.7 ("Authoring Adaptive Hypermedia") we also formulate a design wish for a general-purpose tool — it is desirable that a general-purpose tool provides a generic authoring environment offering a user-friendly interface for designing various types of applications.

2.9 Summary of AHS

The overview of the adaptive hypermedia field shows that many systems target a specific application area and, as a consequence, they cannot be reused in other domains. Up to now we did not manage to find an example of a system that could be considered a truly general-purpose tool. In this chapter we started addressing research question 1 of this dissertation "Is an AHS necessarily a special-purpose tool?". Following the AHAM reference model we derived a set of design requirements a general-purpose tool should satisfy as follows:

1. design requirement "adaptation types and techniques" — provide adaptation of content, links and layout and implementing techniques as described in the Brusilovsky taxonomy [Brusilovsky, 2001] (see figure 2.1).

2. design requirement "aspects used for adaptation" — perform adaptation based on three types of information — user data, usage data and environment data.

3. design requirement "arbitrary concept relationships" — allow arbitrarily many types of concept relationships.

4. design requirement "generic adaptation rules" — allow a "generic" way of defining adaptation rules.

5. design requirement "handling non-monotonic user model updates and cycles in adaptation rules" — handle *non-monotonic* user model updates and *cycles* in adaptation rules.

Furthermore a general-purpose system should provide a reasonable performance and an authoring environment for designing various types of applications.

An answer to research question 1 can be found by designing and implementing a system that would address the above mentioned issues. It is obvious that the more general a system should be, the more difficult its development is.

Rather than start everything from scratch we try to build such a system based on the existing AHA! system version 1.0. We present the implementation results as well as an answer to research question 1 in the following chapter.

Chapter 3

The AHA! System: Version 3.0

This chapter presents the (design and development of the) AHA! system, the Adaptive Hypermedia Architecture, up to and including version 3.0. It describes the rationale for AHA! features that were introduced since AHA! version 1.0, changes which brought AHA! much closer to the AHAM model presented in the previous chapter. This chapter provides a basis for the following chapters in order to understand how further implementation of learning styles became possible in AHA! 3.0.

Section 3.1 describes the history of the AHA! system, its overall architecture, adaptation types and techniques provided by AHA! and the user interaction with the system. Section 3.2 presents different AHA! components in relation to corresponding AHAM components. Section 3.3 gives an overview of the AHA! authoring tools. Section 3.4 describes how AHA! achieves its good performance (for a Web application). Throughout these sections we look back at the list of the design requirements for providing a general-purpose tool presented in chapter 2 and check if they were addressed in AHA! 3.0. Section 3.5 summarizes the chapter and provides an answer to research question 1 of this dissertation "Is an AHS necessarily a special-purpose tool?".

3.1 AHA! System

3.1.1 History of AHA!

At the Eindhoven University of Technology the development of the AHA! system dates back to 1996, when an on-line course text on the subject of hypermedia was augmented with adaptive content and linking. Since then the software for that course has been changed and extended, which led to AHA! version 1.0 [De Bra et al., 2000; De Bra and Ruiter, 2001]. This first "public" version has been studied and experimented with by several research groups in different countries [Cini and de Lima, 2002; Calvi and Cristea, 2002; Romero et al., 2002].

Through a grant of the NLnet Foundation the system was extended with a number of features introduced in the AHAM model [Wu, 2002], such as event-condition-action rules, concept relationships, described through generic rules (see sections 3.2.1.1, 3.2.1.2 and 3.2.1.3), a more flexible user model structure allowing multiple concepts, and a more versatile structure of these concepts with arbitrary attributes (see section 3.2.2). This version of the system, named AHA! 2.0, was developed in parallel with a number of authoring tools to support the process of creating adaptive applications. AHA! 2.0 was presented in a number of publications [De Bra et al., 2002a,b,c; Stash and De Bra, 2003].

The system was developed further on towards AHA! version 3.0. It provides a new, more efficient way of handling conditional content on the pages through the use of conditional objects as an alternative to conditional (inline) fragment inclusion (see section 3.2.3); ways of stabilizing presentations, layout capabilities, and more extended versions of the authoring tools (see section 3.3). AHA! 3.0 was presented in [De Bra et al., 2003a, 2004b,c, 2006] and studied in [Primus, 2005].

In this dissertation we refer to this latter version 3.0 unless stated otherwise. The description in this thesis does not aim at completeness but at providing sufficient understanding of the functionality of and rationale behind AHA! to prepare for the later chapters on the realization of learning style support in AHA!.

3.1.2 AHA! architecture

The overall AHA! architecture is presented in figure 3.1.

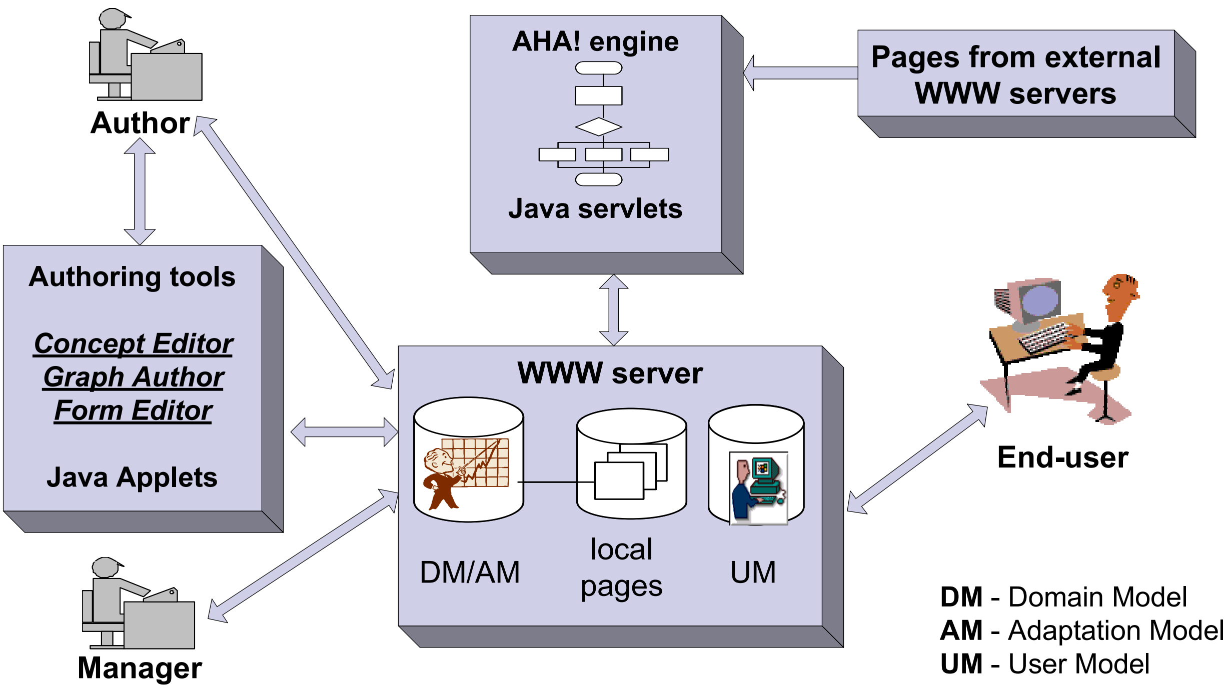


Figure 3.1: Overall AHA! architecture

AHA! is an adaptive webserver extension. The adaptation engine consists of Java servlets that are activated when the webserver receives HTTP requests from the browser (typically when the user requests a page). AHA! can present and adapt webpages from *local* and *remote* webservers.

The following users interact with the system:

- A *Manager* — system administrator — registers "authors" who want to create their adaptive applications using AHA!. (S)he also performs the initial AHA! configuration.

- An *Author* — AHA! author — creates the structure of an application (e. g., a course) consisting of concepts and adaptation rules. An author can design his/her application using one of the authoring tools — the low-level "Concept Editor" or the high-level "Graph Author" (described in more detail in section 3.3). Besides this structure the author also has to create (write or import) the application content, usually consisting of a set of XHTML[1] pages. AHA! does not provide a specific authoring tool for creating pages as the author can use any text or XML (XHTML) editor.

  Using the "Form Editor" tool the author can create forms through which users may update their user model.

  The layout capabilities of AHA! allow the author to provide different layouts for different types of information, such as information pages or glossary; to display a (partial) table of contents and information about concepts (parents, siblings, children); to organize information about the application in various multi-frame structures. The author can thus design various presentation styles (also called *skins*) for his/her applications.

- A *User* — end-user of an application — registers with an AHA! application, and receives a personalized browsing experience.

The combined domain/adaptation model (more about this in 3.2.1) and the user models (for all users) can be stored as XML files on the server, or in a mySQL[2] database. The manager chooses between file and database storage. AHA! can convert the files from XML files to mySQL and vice versa (but no users must be logged on at that time).

3.1.3 Adaptation types and techniques provided by AHA!

In this sub-section we look at the design requirement "adaptation types and techniques" (section 2.8) and check whether and how it is implemented in AHA!.

AHA! provides two kinds of adaptation: *adaptive presentation* and *adaptive navigation support*. Concerning *layout adaptation*, AHA! allows for designing various layouts (presentation styles or skins) and applying them to the whole presentation. Thus the same

[1]http://www.w3.org/MarkUp/
[2]http://www.mysql.com/

application can work with different presentation styles. However, once an author applies a certain style to the application the layout is further not adapted to the user during his/her interaction with the system.

We discuss below the adaptation techniques applied in AHA! for providing adaptive presentation and adaptive navigation support and summarize the results in figure 3.2.

Adaptive presentation

The chosen approach to providing adaptive presentation (or content adaptation) in AHA! is to offer (almost) only the *inserting/removing fragments* technique, which falls into the *canned text adaptation* category in Brusilovsky's taxonomy [Brusilovsky, 2001]. Any arbitrary condition over user model information can be used to decide on the inclusion or exclusion of a fragment.

Using this low level technique a number of other adaptation techniques can be implemented as well. An object in AHA! can have various representations under different conditions. By conditionally including a certain representation of an object in a page, AHA! provides the *altering fragments* technique. The *stretchtext* technique can be realized in AHA! through conditional fragments only when they are used to conditionally include pieces of content, and not to provide alternatives to fragments. Stretchtext also implies that the user can open and close fragments at will. Realizing this without loading entire pages (with alternative content) requires the use of AJAX[3] (Asynchronous JavaScript and XML) technology, which is possible in AHA! because AHA! can process arbitrary XML data.

AHA! also provides multimedia adaptation by using the SMIL or (X)HTML + SMIL [De Bra and Stash, 2004] formats. This realizes the *adaptive multimedia presentations* and *adaptation of modality* techniques.

One form of content adaptation is not realized through *inserting/removing fragments*: the presentation of fragments in a different style (emphasized or deemphasized in some way, e. g., by changing font, size or background) [Ansems, 2002], resulting in a generalization of what Brusilovsky calls the *dimming fragments* technique.

There are only two adaptive presentation methods/techniques in Brusilovsky's taxonomy that are not really possible with AHA!'s simple low level content adaptation techniques: AHA! does not provide the adaptive *sorting of fragments* and does not provide any *natural language adaptation* techniques.

Adaptive navigation support

As with adaptive presentation the basic approach in for adaptive navigation support in AHA! is to have one powerful adaptation technique that can be used to simulate/implement many other techniques mentioned in Brusilovsky's taxonomy. The main technique applied in AHA! for adaptive navigation support (or link adaptation) consists of three parts:

[3] http://en.wikipedia.org/wiki/Ajax_(programming)

- The link anchor tag can be adaptively assigned a *class* attribute, based on arbitrary conditions over the user model (UM). A *style sheet* defines which color corresponds to which link class (and whether the link anchor will be underlined or not).

- An arbitrary number of icons (images) can be presented in front of of or behind the link anchor, each selected based on arbitrary conditions over UM.

- The link destination is typically the name of a concept from DM. (It can also be a URL of a page, in which case the AHA! engine first finds the corresponding concept from the domain model (DM).) Based on arbitrary conditions over UM a resource (page) is selected to be the actual link destination. Having conceptual link destinations that are adaptively translated into "real" (resource) destinations is an adaptation technique not yet presented in Brusilovsky's taxonomy.

A typical use of this link adaptation technique in AHA! applications is to choose three link classes, based on the "desirability" and "visited" status of the link destination (concept):

- *good* — meaning desirable and not previously visited,

- *neutral* — meaning desirable but previously visited, and

- *bad* — meaning not desirable.

AHA! applications can use a different link adaptation in each "view" of the presentation layout. Often an AHA! application will present a kind of navigation menu in which the InterBook [Brusilovsky et al., 1998] style is followed by using colored balls (green, white, red) in front of the link anchors, whereas in the presented pages only the link anchors themselves are shown (without icons), in adaptively selected colors.

Since AHA! 2.0 an extension to XHTML was introduced of which we shall describe some details later. One aspect of this extension is that when using it AHA! can generate the stylesheet for the link colors automatically. In AHA! 2.0 the default color scheme was fixed, but in AHA! 3.0 the author can include a (reference to a) stylesheet defining the color scheme.

Whereas the author of an application defines this link color scheme (with the default colors for good, neutral and bad links) the end-user can opt to override the author's choice and substitute the default colors for diferent ones, by means of a special color selection form. Depending on the choice of colors AHA! can be configured to use:

- the *link hiding* technique with undesirable links shown in the same color as the normal text and not underlined. The user can then only see that there is a link if (s)he moves a mouse over the link anchor.

- the *link annotation* technique with all links shown in visible, different colors.

The default color scheme is *blue*, *purple* and *black* (for good, neutral, bad). This color scheme is used in most existing AHA! applications, like the on-line AHA! tutorial[4], the hypermedia course 2L690[5] offered by TU/e and the on-line adaptive research paper[6] on the design of AHA! [De Bra et al., 2006].

The *link removal* and *link disabling* techniques from Brusilovsky's taxonomy can be simulated in AHA! through conditional content. In the first case the whole link anchor (text) and in the second case only the link anchor tag (and not the text) are conditionally inserted/removed.

The *direct guidance* technique can be achieved in AHA! through the adaptive link destination selection. However, all the possible link destinations and the conditions to decide which destination to use must be defined by the author, so this is a laborious way to perform direct guidance.

Despite the use of one powerful low level link adaptation technique there are some adaptation methods from Brusilovsky's taxonomy that cannot be (easily) implemented using the one AHA! technique: AHA! does not provide *link sorting*, *link generation* and *map adaptation*.

Note that the *adaptive link destinations* technique AHA! offers (and that is not present in Brusilovsky's taxonomy) is actually not so clearly a case of *adaptive navigation support*. On the one hand, this technique manipulates links, which makes it a form of adaptive navigation support. But on the other hand, one can also consider it an extreme form of content adaptation, where the adaptation does not conditionally include a fragment or an object but a whole page. And in fact, when we turn to the discussion of the conditional inclusion of objects we shall see that AHA! uses a single *adaptive resource selection* mechanism for both the conditional inclusion of objects (clearly a form of adaptive presentation) and for the adaptive link destinations (a form of adaptive navigation support).

To summarize this sub-section, we list the adaptation techniques and their possible presence in AHA! in figure 3.2.

3.1.4 Interaction between an end-user and AHA!

Figure 3.3 shows how AHA! generates an adaptive page for a user.

A user can make a request for a concept (from DM) or a page. The system decides on their presentation to the user based on the information from UM. When the user requests a concept, one of the resource pages associated with it is selected (using *adaptive resource selection*. Further on, the selected page is processed in the same way as the page requested directly:

- static pieces of information (unconditional fragments) are inserted in the output page unchanged;

[4]http://aha.win.tue.nl/tutorial/
[5]http://wwwis.win.tue.nl/2L690/
[6]http://aha.win.tue.nl/ahadesign/

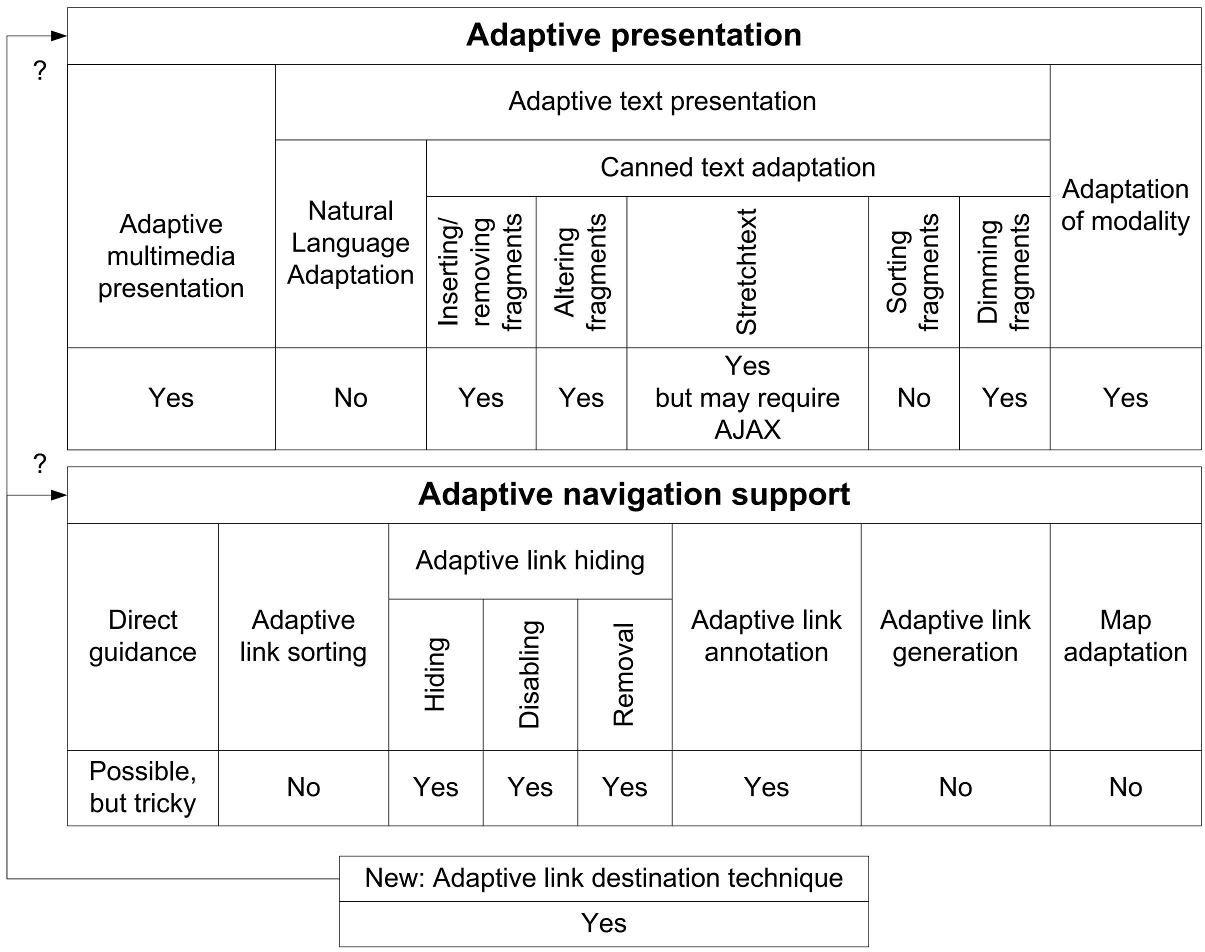


Figure 3.2: Adaptation types and technologies provided by AHA!

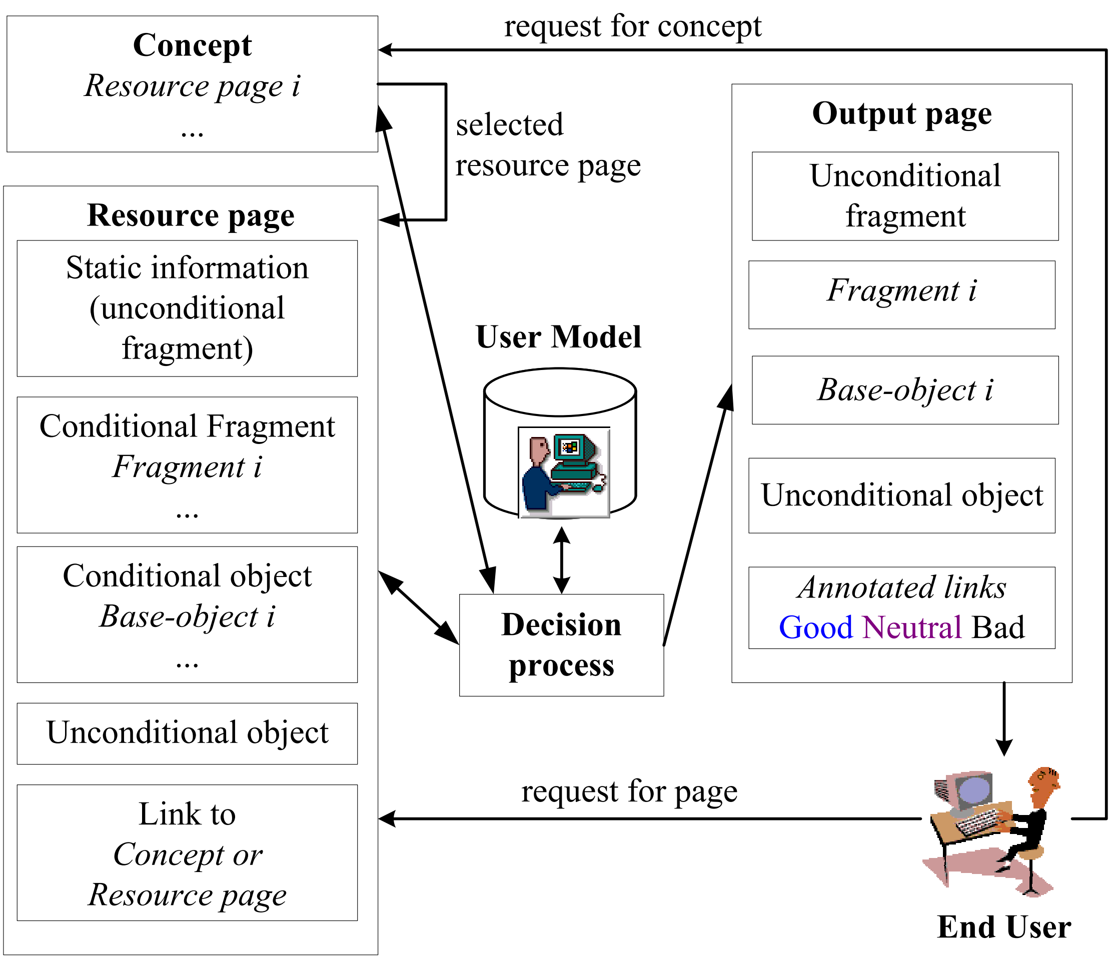


Figure 3.3: Processing an AHA! page

- conditional fragments (pieces of information written in the resource page) are shown/ hidden under various conditions, and there may be fragments within fragments as well;

- for (external) conditional objects some of the *base-objects* (fragments, specified in separate files) representing them under various conditions are included (using *adaptive resource selection*, possibly recursively including other objects;

- pages may also include unconditional objects such as, for instance, the header in the AHA! tutorial, containing links to servlets that generate pages containing information about what the user has already seen and has not seen yet from the application domain, forms to change various settings in the user model, etc.;

- conditional links are translated to *good*, *neutral* or *bad* depending on their desirability and visited status, and if specified in the layout some icons may also be added to the link anchors.

AHA! uses *event-condition-action rules* to perform UM updates. These rules are triggered when the user accesses a concept or a page. As a consequence, if there are conditional objects on the (selected) page, the rules associated with an access to these objects are also triggered. Therefore the sequence of actions performed by the system when the user accesses a concept/page is as follows:

1. The user accesses a concept/page.

2. UM is updated, based on that access. If the user accessed a concept, one of the resource pages associated with it is selected.

3. While parsing the (XML or XHTML) page conditional objects may be included. The access to an included object causes further UM updates.

4. The object's (re)presentation is determined based on the updated state of UM.

5. The process from step 3 is repeated, if the object includes other objects, until there are no more objects to process. A badly designed application could have objects that include themselves (directly or indirectly) in an infinite recursive loop. AHA! limits the number of object inclusions to abort such a loop.

6. The adapted page with its objects (and conditionally included fragments) and annotated links is sent to the browser.

3.1.5 Summary of the AHA! description

In this section we presented a general overview of the AHA! system — the history of its development, the overall architecture, the provided adaptation types and techniques and the end-user interaction with the system.

Rather than start building a general-purpose system from scratch we have built the current AHA! version 3.0 based on the educational adaptive hypermedia engine of AHA! version 1.0. In section 3.1.3 we looked at the design requirement "adaptation types and techniques" for providing a general-purpose tool. Not all adaptation types and techniques that were summarized in 2.8 are provided by AHA! yet. For example, AHA! does not provide layout adaptation, stretchtext technique cannot be directly realized in AHA! but can be obtained by using AJAX, direct guidance is possible but tricky and AHA! does not provide natural language adaptation, sorting fragments, adaptive link sorting and adaptive link generation techniques. On the other hand, AHA! 3.0 adds a novel technique called "adaptive link destinations" that is not specified in the Brusilovsky taxonomy [Brusilovsky, 2001] yet and that is in fact an application of *adaptive resource selection* that is used for the conditional inclusion of objects as well. The other techniques that did not exist in AHA! 1.0 but were added in AHA! 3.0 enable adaptive multimedia presentation, adaptation of modality and dimming fragments.

3.2 AHA! in Relation to the AHAM Model

In this section we show different AHA! components, such as domain, adaptation and user models (DM, AM and UM) and AHA!'s within-components layer in more detail and compare them to corresponding AHAM model components. A first attempt at this was done in Wu [2002], for AHA! version 1.0. We will often refer back to this description, and add updates that are related to AHA! versions 2.0 and 3.0 where needed.

3.2.1 AHA! Domain/Adaptation Model

The most obvious way in which AHA! deviates from the AHAM reference model is by not separating DM from AM. In AHA! *concepts* are defined along with *requirements* that determine under which conditions the user is "ready" to access these concepts, and *generate rules* that specify how the browsing behavior of the user translates into user model updates. The *generate rules* are *event-condition-action rules* as used in active database theory [Baralis and Widom, 2000]. However, non-separation of the domain and adaptation models is just an implementation issue. The models are combined at the low level used by the AHA! engine.

In order to make a clear separation between the domain and adaptation models, as proposed by the AHAM model, a high-level authoring tool Graph Author was developed (see also 3.3.1). Through the Graph Author the definition of concepts and concept relationships comprising DM and adaptation rules comprising AM becomes separate. These issues are discussed in the following sub-sections. When "saving" a DM in the Graph Author, the translation of the concept relationships into adaptation rules for the combined DM/AM the AHA! engine needs is performed, based on templates. The results of this translation can then be used by the AHA! engine.

As was mentioned in 3.1.2, the combined DM/AM can be stored in XML files on a server or in a mySQL database. The XML language was chosen for a number of reasons: it is a W3C[7] recommendation, the language was designed to describe, structure, store and exchange data, and furthermore the language is extensible allowing for the definition of arbitrary tags to describe that data. At the time the AHA! project (or Adaptive Hypermedia for All) was started, in 2001, XML was getting an increasing popularity and up to now it remains one of the most common tools for data manipulation and data transmission. Another format for storing of the DM/AM of AHA!, the mySQL database, was provided to ensure scalability.

3.2.1.1 Concepts

The AHAM model considers three levels of concepts: *abstract concepts*, *pages* and *fragments*. The lowest level concepts in the DM of AHA! 1.0 and 2.0 are *pages*. Conditionally included fragments are not represented in DM. In AHA! 3.0 the lowest level are the conditionally included *objects*. These correspond to the *fragments* of AHAM. AHA! 3.0 still allows conditionally included fragments that are not represented in DM too.

In AHAM, as described in [Wu, 2002], it is assumed that fragments are "atomic" within the model, which means that fragments cannot contain other fragments. This is the only AHAM assumption that does not hold in AHA! because conditionally included objects may (conditionally) include other objects. Note that when an object A, with condition C, includes another object B, with condition D, it looks like we can "flatten" the conditional inclusion. Let A1 be the part of A before the inclusion of B, and A2 be the part of A after the inclusion. We can then replace the inclusion of A by a conditional inclusion of A1 with condition C, then the conditional inclusion of B with condition "C and D", and then the conditional inclusion of A2 with condition C. There are two problems with this scheme: when A1 is included, user model updates are performed that might cause the second evaluation of C, as part of "C and D" in order to decide on the inclusion of B to have a different outcome. Likewise, after including B the condition C used to decide on the inclusion of A2 may have a different value than the first time, because of the updates caused by including A1 and the updates caused by including B. So flattening does not really work. The conditional object inclusion in AHA! really is more powerful than the *fragments* of the AHAM model.

In AHA!, different concept types can be defined using *concept templates* (that are specified as XML files). Every AHA! concept has a name and optional description, and is of a certain *type* (e. g., abstract, object, page), represented by a template. Following the AHAM model, since AHA! 2.0 the system was extended with the ability of adding arbitrary attributes to concepts [De Bra et al., 2002a]. The template determines which attributes the concept has, and whether it must have a resource (file) associated with it or not. In figure 3.4 we show an example (in pseudo-code) of a *page concept template*.

[7] http://www.w3.org/

```
Template
Name: page concept
Attributes:
    attribute name: access
    description: triggered by page access
    default: false
    type: bool
    isPersistent: false
    isSystem: true
    isChangeable: false

    attribute name: knowledge
    description: knowledge about this concept
    Default: 0
    type: int
    isPersistent: true
    isSystem: false
    isChangeable: true

    attribute name: visited
    description: has this page been visited?
    default: 0
    type: int
    isPersistent: true
    isSystem: true
    isChangeable: false

    attribute name: suitability
    description: the suitability of this page
    default: true
    type: bool
    isPersistent: false
    isSystem: true
    isChangeable: false

Has resource: true
Concepttype: page
Conceptrelations:
    concept relation name: knowledge_update
    label: 35
```

Figure 3.4: Page concept template

In this example template a page concept has 4 attributes and one unary concept relationship which is the *knowledge update*. This relationship specifies an event-condition-action rule which is described in sub-section 3.2.1.3.

Each attribute of a concept can be described as:

- *system* — having a special meaning for a system,
- *persistent* — meaning that the value is stored (permanently, but can be updated),
- *changeable* — meaning that it can be used in a form that lets end-users change the attribute value.

The attributes of a page concept usually play the following roles (we say "usually" because attributes that are not system-defined can play any role we want):

- *access* — when the page is accessed the rules associated with this attribute are executed (first). This is a *system* attribute (indicated by "isSystem: true" that only serves the purpose of starting the adaptation rule engine. The value of this attribute is not stored in UM ("isPersistent: false").
- *knowledge* — this attribute is typical for educational applications. It stores the system's idea of the user's knowledge of the concept. The knowledge attribute has no special meaning to the AHA! system. It has "isSystem: false" to indicate this. It stores an integer value in UM (as can be seen from the "isPersistent: true" property and from the "type: int". Initially the user's knowledge of every concept is set to 0 ("default: 0"). By default the user is allowed to change this value through the special forms created with the Form Editor, because of "isChangeable: true".
- *visited* — this attribute can be used by the system to remember (hence it has "isPersistent: true") whether the user visited the page or not. Visited can be defined as a Boolean (visited or not visited) or as an integer (to count the number of visits). In figure 3.4 it is an integer. With the default link adaptation rules (which can be changed by the author) this attribute is used by the system to decide whether a suitable link will be displayed using the Good (blue) or Neutral (purple) color, to calculate the number of visited concepts and to produce a list of visited concepts. In the example, the system will not allow end-users to change the value of this attribute through a form ("isChangeable: false"), but this too is just an example. AHA! does not forbid the creation of a template in which visited is made changeable.
  The default *knowledge update* rule (described in 3.2.1.3) will set the value of *visited* (as well as the knowledge attribute). It will only register the page as visited if it was visited when *suitable*.
- *suitability* — this attribute can be used by the system (with the default link adaptation rules) to decide on the presentation of links on the page. If the suitability is true the link anchors will be Good (blue) or Neutral (purple) depending on the visited

status. If the suitability is false the anchors will be Bad (black). In the example template *suitability* is defined as a non-persistent attribute, which means that its value is calculated from the *default* each time it is needed. This default is set to *true* in the template. If there are prerequisites for the concept they will result in conditions that are added to this default value.

Concepts can have more aspects to be controlled by AHA! and that can be added using the authoring tools, namely *stability*, *visibility* and *adaptive resource selection*.

Stability is used for keeping the presentation of a concept "stable". It means that once the concept has been presented it is always presented in the same way, even if the UM instance would suggest otherwise. As already mentioned in section 2.1, users may find it disturbing when a presentation changes on them. Stable presentations alleviate this problem. The stability can be chosen to be forever after the first adaptation, stability during the current session, or stability while a Boolean expression remains true.

When a concept is defined using a template that has a *showability*[8] attribute (or when a showability attribute is added) the resource associated with the concept can be selected based on expressions. When a concept is accessed the resource that will be presented to the user is determined by examining expressions. This *adaptive resource selection* construct can be used for deciding which page to present when following a link to a concept (as in *adaptive link destinations* or which resource to include when accessing a page *conditionally included objects*.

When a concept is defined using a template that has a *visibility* attribute (or when a visibility attribute is added) fragments using a <span> tag with the class "conditional" and the id of that concept can be presented in a different style (emphasized or deemphasized in some way) depending on the value of the visibility attribute.

Authors can create their own concept templates or add attributes to existing ones. Apart from the system-defined attributes an arbitrary number of other attributes may be added and used in adaptation rules in any way the author likes.

3.2.1.2 Concept relationships

In order to connect concepts in AHA! 1.0 using AHAM constructs a number of concept relationship types were defined in [Wu, 2002]. These relationships are not really present in AHA! but they were used to show how concepts were *implicitly* related in AHA! 1.0:

- *Fragment(P; F)* means F is a fragment of page P. Through the *Fragment* relationship we can decide on the inclusion/exclusion of fragments of a page.

- *Link(P1; P2)* means that there is a hyperlink from P1 to P2. Through the *Link* relationship we can perform adaptation of link anchors based on the desirability of the link destination.

[8]Actually, any attribute can be used for this purpose, but the default templates suggest to use this attribute name.

- *Contains(C; P)* means that the abstract concept C "contains" page or concept P. The *Contains* relationship is used to be able to perform an action on user model attributes of all pages of an application (or course) at once.
- $Contains^{-1}(P, C)$ is the inverse of *Contains*. We only use this relationship type to find out to which application (or course) a page belongs.

In a similar way we can extend this set of constructs to include new relationships for the AHA! 3.0 features of conditionally included objects and of links to concepts (instead of to pages):

- *Object(P; O)* means O is an object of page P. Through the *Object* relationship we can decide on the inclusion of objects in a page (similarly to the previous fragment inclusion).
- *Link(P; C)* means that there is a hyperlink from a page P to concept C.

AHA! version 1.0 only offered authors the ability to write *specific* adaptation rules — rules tied to concrete concepts only. Therefore it was not possible to define concept relationship types such as prerequisites. AHA! 3.0 shows a much closer resemblance to AHAM by allowing *generic* adaptation rules based on relationship types, and implemented through templates. Generic adaptation rules deal with "generic" concepts, therefore they do not need to be replicated for every concept they apply to. This corresponds to design requirement "generic adaptation rules" for creating a general-purpose tool (see section 2.8).

Furthermore, as already proposed in AHAM, AHA! authors can (in theory) define arbitrarily many types of relationships. This fact meets the design requirement "arbitrary concept relationships" for creating a general-purpose tool (see section 2.8).

In [De Bra et al., 2002] the authors describe how the notion of concept relationship types can be incorporated into the AHA! system (using an XML notation). The proposed high-level authoring tool Graph Author made creating an adaptive application as simple as drawing a (labeled) graph of concept relationships of various types.

Concept relationship types are defined using templates, just like concepts. The definition consists of two parts (actually defining 2 XML files):

- *a presentation part* — it is used by the Graph Author to decide how to show relationships of this type (color, arrow style) and whether the relationship type must be acyclic. (A relationship type marked as "acyclic" has the property that the graph of all the instances of this type must be acyclic. For prerequisites, for instance, it means that no concept may be a prerequisite for itself, directly or indirectly. Acyclic relationship types may therefore be used to define trees or forests.)
- *an implementation part* — used by the Graph Author to translate a conceptual model into the actual adaptation rules used by the AHA! engine.

We present the interface of the Graph Author in section 3.3. Below we give some examples of relationships (only the implementation part) used in the Graph Author to

```
Relation type
name: prerequisite
listitems:
    (setdefault location="destination.suitability" combination="AND"):
    source.knowledge > var:50
```

Figure 3.5: Prerequisite relationship

illustrate their generic character. These definitions of concept relationship types show how using the templates a translation can be made to the event-condition-action rules used by the AHA! engine. The engine itself only knows about concepts, attributes and these rules; it does not know about concept relationships and concept relationship types.

Figure 3.5 shows an example in pseudo-code of how prerequisite relationships are specified in the Graph Author template format. Prerequisites form the most often used type of relationships in various kinds of applications.

The presented example shows that for a "destination" concept to become desirable the knowledge of the "source" concept should be greater than 50 indicated by setting *destination.suitability* to *source.knowledge>var:50*. This value 50 was chosen arbitrarily. The *var* means that the author can specify a different threshold for an instance of the prerequisite relationship. (The Graph Author allows an alternative threshold to be specified as a label for the arrow in the concept relationship graph.)

This example shows that the relationship deals with generic concepts, specified by the two variables *"source"* and *"destination"*. In terms of Graph Author "source" and "destination" are correspondingly concepts from which and to which an arrow indicating a certain relationship is drawn. While saving the graph of concept relationships these names will be replaced with specific concept names. The author can also use specific names directly in the description of the relationship template but this is only useful if a concept relationship type has a fixed relationship with a fixed concept.

Prerequisite relationships are translated into an expression that is used as the *default* value for the *suitability* attribute (of the destination concept). Because that attribute is volatile (not persistent) the expression is evaluated when the value of the attribute is needed. A single concept may have multiple prerequisites. The *combination="AND"* and *combination="OR"* indicate that the expressions for the different prerequisites must be combined with a logical AND and OR operator respectively. This means that *all* or *at least one of* the prerequisites of a concept must be satisfied before the concept will become desirable.

An interesting question is whether prerequisite relationships are *transitive*. In other words, when A is a prerequisite for B and B for C, is A then automatically also a prerequisite for C? The answer is not a clear "yes" or "no". Looking at the definition in figure 3.5 we see that *destination.suitability* only depends on *source.knowledge*, not on *source.suitability*. Adding this requirement would ensure the transitivity. Without this addition the transitivity in principle does not hold. However, if we skip forward a bit and look at figure 3.8

we see another interesting rule: when a non-suitable concept is visited its knowledge value is increased to 35, which is lower than the prerequisite threshold of 50. So if concept A is not known, and B is visited, the knowledge of B does not increase enough to reach the threshold for the prerequisite for C. Thus, when the prerequisite for B is not satisfied the knowledge update is such that the prerequisite for C is still not satisfied, which makes prerequisites behave like they are transitive (even though strictly speaking they are not).

Figure 3.6 shows an example of a domain model (taken from the on-line AHA! 2.0 tutorial) as it would appear in the Graph Author.

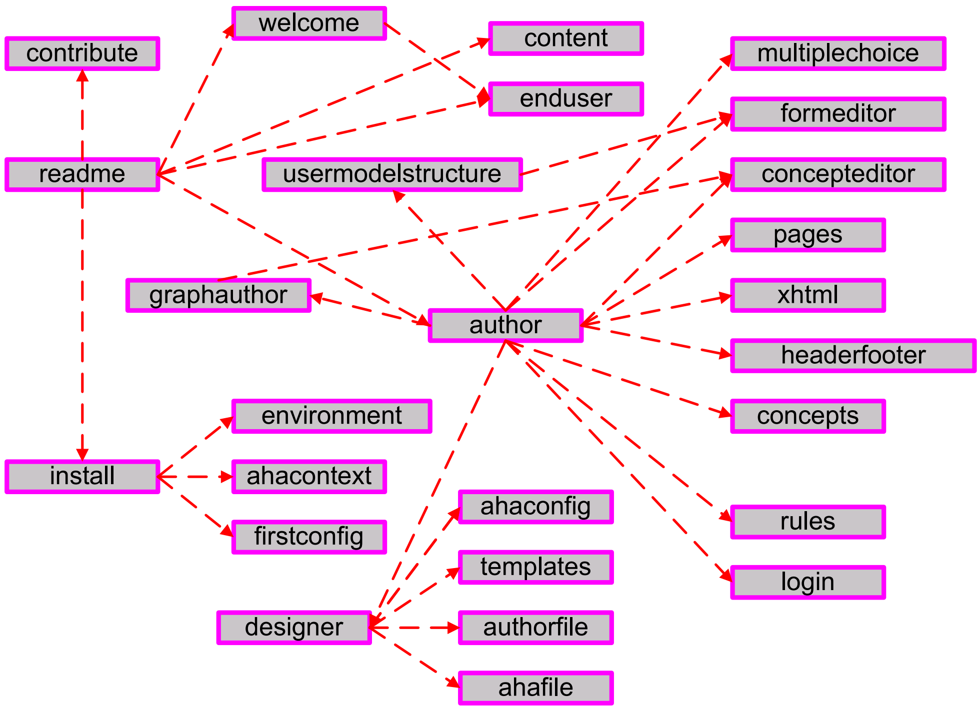


Figure 3.6: Domain model with concepts and prerequisite relationships

A number of concepts in this application are connected through prerequisite relationships, for example, the concept “readme” is a prerequisite for the concept “welcome”. While saving the graph of concept relationships, an expression *tutorial.readme.knowledge>50* will be created as the default for the suitability attribute of *welcome*. The prefix *tutorial* is the name of the application.

As another example of concept relationship types we consider the “inhibitor”, which can describe an unusual type of desirable reading order: concept B inhibits concept A means that after visiting B it is no longer desirable to visit A. The pseudo-code describing this relationship type is presented in figure 3.7.

The presented example shows that the “destination” concept is only desirable when the “source” concept is still unknown (knowledge of the “source” concept is 0).

```
Relation type
name: inhibitor
listitems:
    (setdefault location="destination.suitability" combination="AND"):
    source.knowledge == var:0
```

Figure 3.7: Inhibitor relationship

The templates for other relationship types like the *knowledge update* are more complicated as they define event-condition-action rules that must be generated. We discuss such rules in the next section.

3.2.1.3 Adaptation Rules

The AHAM reference model proposes an adaptation model based on condition-action rules or event-condition-action rules. AHA! 1.0 had very simple rules associated with a single numerical attribute of a concept (called "knowledge") mostly intended to describe the propagation of knowledge from pages to sections and from sections to chapters. AHA! 3.0 tries to preserve its initial simplicity while making the adaptation more versatile.

As has already been mentioned in section 3.2.1.1, every concept in AHA! 3.0 can have an arbitrary number of attributes. With each attribute of each concept an arbitrary number of adaptation rules can be associated. Figure 3.8 shows how such adaptation rules are created from concept relationships. Each rule (called "generateListItem" for historical reasons) is triggered by an update to an attribute of a concept. The attribute is specified as rule's "location". In the example of figure 3.8 this attribute is *source.access*. Note that *access* is a system attribute that is not really "updated" but that triggers the associated rules when the concept is accessed. Each rule has a condition (called "requirement") that is checked to see whether the associated actions should be performed. The adaptation rule may have a number of "trueActions" (executed when the condition is true) and also a number of "falseActions" (executed when the condition is false).

The "knowledge update" relationship shown in figure 3.8 is a bit of a special rule: it only uses the source concept, and not the destination. This is because it is used as a unary relationship: it specifies adaptation rules that involve just a single concept. The relationship specifies two event-condition action rules that should be executed when the (source) concept is accessed (as they are tied to the attribute "access" of the concept). The first rule checks whether the source concept is not suitable and whether its knowledge is lower than 35. This value 35 was chosen arbitrarily. If this is the case the knowledge is set (increased) to 35. If the concept is suitable or if its knowledge is already 35 or more then this rule does nothing. The second rule checks whether the source concept is suitable, and if so it sets the knowledge attribute to 100, and it sets the visited attribute to the visited attribute + 1, thus increasing the visited "count" by 1.

```
Relation type
name: knowledge_update
listitems:
    generateListItem isPropagating="true" location="source.access"
    requirement: !source.suitability && source.knowledge < var:35
    trueActions
        action
            conceptName: source
            attributeName: knowledge
            expression: 35
    generateListItem isPropagating="true" location="source.access"
    requirement: source.suitability
    trueActions
        action
            conceptName: source
            attributeName: knowledge
            expression: 100
        action
            conceptName: source
            attributeName: visited
            expression: source.visited + 1
```

Figure 3.8: Knowledge update relationship

Figure 3.9 shows in pseudo-code the result of the translation of the knowledge update relationship applied to "readme" concept of the "tutorial" application as it appears in the "AHA! authoring format" used by the Concept Editor and the Graph Author.

Another relationship type widely used in educational applications is *knowledge propagation*. When the knowledge of a concept changes that change is propagated to the concepts that are higher in the concept hierarchy. How much knowledge is propagated depends on the number of siblings the concept has. The idea is that when all siblings reach a knowledge level of 100 the parent should reach 100 as well. (However, due to integer arithmetic and truncation that value may end up being slightly lower.) Figure 3.10 shows the definition of knowledge propagation relationship type.

The knowledge propagation relationship has two generic variables "child" and "parent". These are synonyms for "source" and "destination", but easier to understand when the relationship follows the concept hierarchy. "Child" and "parent" will be replaced by the names of specific concepts they apply to, using parent-child relationships from the concept hierarchy. The knowledge propagation rule is tied to the "knowledge" attribute of a "child" concept, so whenever this attribute value changes the rule is triggered.

The action of this rule updates the "knowledge" attribute of the "parent" concept if that values is less than 100. The *change* in the knowledge value of the "child" concept is the basis for the update (and not the absolute knowledge value). This is indicated by the prefix "_" (note that _child.knowledge is used instead of child.knowledge). When several knowledge propagation relationships have the same destination (or parent) the fraction of

```
generateListItem isPropagating="true"
requirement: !tutorial.readme.suitability &&
             tutorial.readme.knowledge < 35
    trueActions
        action
            conceptName: tutorial.readme
            attributeName: knowledge
            expression: 35
generateListItem isPropagating="true"
requirement: tutorial.readme.suitability
    trueActions
        action
            conceptName: tutorial.readme
            attributeName: knowledge
            expression: 100
generateListItem isPropagating="true"
requirement: tutorial.readme.suitability
    trueActions
        action
            conceptName: tutorial.readme
            attributeName: visited
            expression: tutorial.readme.visited+1
```

Figure 3.9: Example of translation of knowledge update relationship

```
Relation type
name: knowledge_propagation
listitems:
    generateListItem isPropagating="true" location="child.knowledge"
    requirement: parent.knowledge < 100
    trueActions
        action combination="DIV_S"
            conceptName: parent
            attributeName: knowledge
            expression: parent.knowledge + (var:DIVIDE * _child.knowledge)
```

Figure 3.10: Knowledge propagation relationship

```
generateListItem isPropagating="true"
requirement: tutorial.tutorial.knowledge < 100
    trueActions
        action
            conceptName: tutorial.tutorial
            attributeName: knowledge
            expression: tutorial.tutorial.knowledge +
                        (0.1 * _tutorial.welcome.knowledge)
```

Figure 3.11: Example of translation of knowledge propagation relationship

the knowledge value change that is actually propagated is calculated by dividing 1.0 by the number of sources (children). The term "DIV_S" in figure 3.10 indicates division by source of the relationship. "DIV_D" would mean division by destination. The divider can be calculated by default but it can also be overridden by the label in the graph, allowed because of "var:DIVIDE". Through these values we can show that knowledge contribution can be distributed unevenly.

Figure 3.11 shows how the rules associated with the knowledge attribute of a "welcome" concept are translated. The concept "tutorial" has 10 children which contribute to its knowledge. "Welcome" is one of them. Therefore 0.1 of its knowledge change is propagated to a "tutorial".

The adaptation engine keeps track of which attributes of which pages or concepts are updated by rule actions, and "triggers" the execution of their associated rules. This process continues until there are no more rules to execute. There is no limit to the number of (condition-action) rules that can be associated with an attribute of a concept.

It is possible to specify rules that would decrease the attribute value of a certain concept upon an attribute value update of another concept. Thus AHA! allows for non-monotonic updates as specified in the design requirement "handling non-monotonic user model updates and cycles in adaptation rules" (see section 2.8).

AHA!'s rule system is powerful, especially because rules can trigger each other (indefinitely). However, in [Wu, 2002] it was already shown that this causes (potential) problems with termination and confluence. *Termination* means that the system provides an adaptation result in a finite number of steps, and *confluence* means that the system provides deterministic adaptation results (independent of the order in which simultaneously triggered rules are executed).

AHA! provides 3 solutions for these problems as described below and therefore allows for handling cycles in adaptation rules as specified in the design requirement "handling non-monotonic user model updates and cycles in adaptation rules" (see section 2.8).

First, an option for the author to foresee the potential problems is having a "propagate" attribute per rule or action that indicates whether this action is allowed to trigger other actions or not. This was also proposed in the AHAM model. As the action indicates which attribute of which concept is updated, it is immediately clear which rule propagation will occur. If the author sees that there is a danger that the rule execution through this propagation mechanism might not terminate (s)he can set the "propagate" attribute to "false" (otherwise to "true"). Furthermore, the author can control more precisely the conditions under which a rule is allowed to be executed. Confluence is not a real problem as the author can control the order of rules execution in the low-level authoring tool Concept Editor provided by AHA!.

Second, the AHA! authoring tools have been extended with the possibility to check for cycles in the applications with various concept relationships. Although generally cycles are allowed, some kinds of cycles may cause the adaptation engine to enter an infinite loop of user model updates. This check is performed automatically when an author creates concepts and relationships but it can be useful to select an option to check for cycles manually when the creation of an application is completed, or when an application is imported from other authoring tools (that may not have been checked for potential infinite loops).

As a third solution, AHA limits the number of objects included in a page. If a recursive objects' structure results in an infinite loop AHA! stops when the number of objects included in a page is 500 showing the page with all recursions up to that number. This is a very simple method for termination but this also means that even if there is no infinite loop the number of objects considered for inclusion into a page should be not more than 500. Though this is an effective way of stopping the recursion this is not really a "correct" one (because it probably does not produce the page the author intended).

Adaptation rules execution scheme

As already mentioned in 3.1.4 in AHA! all the adaptation rules are executed before the page is sent to the browser. This fact is something an author really needs to take into account when designing an application. When the end-user accesses a concept or page, that page can be adapted only once: before the user starts reading it. The http request/response paradigm implies that after the server's response, which is sending a page to the browser, the server cannot send updates to the page, for instance, to take into account the user's progress while reading the page. This is not a peculiarity of AHA! but rather the common behavior of most other Web-based adaptive systems.

An example of what this execution scheme implies is as follows: when accessing a page, the "visited" attribute of the corresponding concept is updated (incremented by 1). This results in links to the page being displayed in purple (neutral) instead of blue (good). If the page contains a link to itself (or to its concept) that link will always be presented as a "visited" link (meaning, in purple), even on the first visit. And when concept A is (the only) prerequisite for B, and page A contains a link to B, the knowledge of A will be updated first, which will result in the prerequisite being satisfied, and as a result the link to B will be "recommended" (shown in the good or blue color). This is the behavior we expect: the link should be recommended after reading page A, and the only way to make it recommended after reading A is to already make it recommended when generating page A.

An example of where this execution scheme might cause a problem is when a page contains a fragment that should be shown when the corresponding concept is not yet known. Making the fragment conditional with the condition that "page.knowledge<50", for instance, will cause the fragment to not be shown at all: the knowledge of the page is updated first, meaning it becomes 100, and the condition for the fragment will never be satisfied. A solution to this is to make the fragment depend on the visited counter. The condition "page.visited $\leq$ 1" will ensure that the fragment is shown the first time (because visited, being 0 initially, is incremented to 1 upon the first "suitable" visit) and only the first time.

For the inclusion of objects the situation is different, because the author can make (clever) use of the known execution order of adaptation rules. Whereas the AHAM model makes no assumption on the execution order for rules that are triggered simultaneously, the AHA! system has a more deterministic behavior: rules are executed in the order in which they appear in the templates, or in the concrete concept definitions in DM/AM. We give a small example of the adaptation rules for object inclusion in pseudo-code below (in figure 3.12), for the conditional inclusion of an explanation of a concept *au bain marie* in a cookbook (see [De Bra et al., 2003a]).

The execution order of the rules is as follows:

1. When the concept *ahacook.au_bain_marie* is accessed the first action to be executed will set the "showability" to a value that depends on the "knowledge" value for this concept. This thus uses the *old* knowledge value.

2. Next, the "knowledge" value is incremented by 50 (if it is not already 100 or more).

3. Next, and triggered by the change to the "showability" attribute, a resource (called returnfragment) is chosen.

Because the execution order of these rules is known to be the order in which they are written (under the attribute that triggers them) it is easy to set "showability" to a value that depends on the *old* knowledge value instead of the *new* value. For the conditional inclusion of a prerequisite explanation on the page this was not possible because the conditions for fragments are only evaluated after the user model updates corresponding to the page access are already performed.

Note that figure 3.12 does not show the actual AHA! rule syntax, but that is not important as the authoring tools (Graph Author and Concept Editor) hide the syntax from the author anyway.

3.2.2 AHA! User Model

The AHAM reference model proposes a rich user model, with for each page or higher level concept a set of attribute/value pairs. AHA! 3.0 follows the AHAM model with three exceptions:

```
Concept
name: ahacook.au_bain_marie
expr: true
attribute
    name: access
    action
        if: ahacook.au_bain_marie.knowledge < 50
        then: ahacook.au_bain_marie.showability := 0
    action
        if: ahacook.au_bain_marie.knowledge >= 50
        then: ahacook.au_bain_marie.showability := 50
    action
        if: ahacook.au_bain_marie.knowledge == 100
        then: ahacook.au_bain_marie.showability := 100
    action
        if: ahacook.au_au_bain_marie.knowledge < 100
        then: ahacook.au_bain_marie.knowledge += 50
  ...
attribute
    name: showability
    casegroup
        defaultfragment: au_bain_marie.xhtml
        casevalue
            value: 0
            returnfragment: frag_au_bain_marie_large.xhtml
        casevalue
            value: 50
            returnfragment: frag_au_bain_marie_medium.xhtml
        casevalue
            value: 100
            returnfragment: frag_au_bain_marie_small.xhtml
```

Figure 3.12: Example of an object concept structure

- Attributes (of concepts) in the domain/adaptation model can be defined not to be persistent (e. g., “access”, “suitability” attributes). Any updates to these attributes are lost when the AHA! engine terminates.

- When a user logs in for the first time a special concept called “personal” is created, which is used to store user-related aspects that are not about the subject domain of the application. This information includes a name, email, login-id, password, and any other information the application asks for during the login process. Because this information is represented through a concept it can be used for adaptation. (In AHA! 1.0 the login information was not available for adaptation purposes.) Like all concepts in AHA! it can have arbitrary attributes.

  In AHA! 3.0 the form-based registration procedure for the end-users was complemented by an automatic procedure enabling those users who do not want to register in the system to study anonymously. This was done by using the “cookie” mechanism to maintain a unique ID which is generated for each user when (s)he starts the first session. The unique ID is set as a value of an attribute “id” of a concept “personal” in the user model. Anonymous users do not have a password and they have to carry on their anonymous sessions on the same computer using the same browser.

- AHAM does not mention any restrictions on the data types that can be used in UM. In AHA! attribute values can only be integers, Booleans or strings.

These facts answer the design requirement “aspects used for adaptation” (see section 2.8) for providing adaptation based on various aspects of the relation between the user and the domain model concepts, and adaptation to arbitrary user’s characteristics that are not about the domain model. Not all possible information about the user’s interaction with the adaptive application (referred to as usage data in AH terminology) can be used for adaptation. For example, AHA! does not recognize such actions as scrolling. It only recognizes those actions that are related to http requests, such as following a link.

For providing adaptation to the user environment it is possible to create a concept called “environment” with the attributes indicating what the user’s environment is — e. g., *place*, *time*, *equipment*, etc.. Adaptation can be based on the values of those attributes. However in some cases, e. g., if we wish to provide adaptation to user’s equipment — PDA or computer — the layout of the presentation needs to be changed and that is not yet possible in AHA!.

Another design wish is to define arbitrary data types for the attributes of the concepts. AHA! only allows for attribute values of simple types — integer, boolean and string. AHA! version 3.0 also allows for attribute values that are expressions, which are evaluated on the fly, e. g., the non-persistent “suitability” attribute.

In the same way as the domain/adaptation model, user models can be stored as XML files on a server or in a mySQL database.

3.2.3 AHA! Within-Component Layer

According to the AHAM model the application content is implementation dependent and hidden in the Within-Component Layer. Several document formats have been used in the past for storing the application content in AHA!. The system translates the pages into HTML or XHTML format depending on the format of the source page and sends them to the browser. Let us have a look at the evolution of the format for source pages from the earlier AHA! versions to the latest AHA! 3.0 version. This overview will show that AHA! has evolved from a system, in which the internal format of the data was very closely tied to the way the adaptation worked, to a system in which the data format is (almost) independent of the adaptation engine (as the AHAM model dictates).

3.2.3.1 AHA! version 0

The very first AHA! version[9] used the C-preprocessor (with #if constructs) to realize the conditional inclusion of fragments [Calvi and De Bra, 1997]. This was soon followed by a version[10] that used HTML with comments for items used by the adaptation engine [De Bra and Calvi, 1997, 1998], for instance, an "if" statement to conditionally include fragments.

```
<!-- if definition and history -->
  This part appears if the two "concepts" definition and history
  are both known according to the user model.
<!-- else -->
  If this is not the case then this alternative is presented instead.
<!-- endif -->
```

Links that had to be adapted were specified with class "conditional", like:

```
<A HREF="readme" CLASS="conditional">
```

HTML comments were also used to specify knowledge "generated" by reading a page:

```
<!-- generates xanadu -->
```

would say that the page contains knowledge about "xanadu". Likewise comments were used to specify prerequisite knowledge:

```
<!-- requires xanadu and intermedia -->
```

would mean that the pages about "xanadu" and "intermedia" should be read before links to this page will become recommended.

From this description it is clear that in the earliest AHA! versions both the specification of the content adaptation, of the prerequisite relationships that lead to link adaptation and of the rules for performing the user model updates are all embedded in the pages, meaning that there is no separation between the *Storage Layer* and the *Within-Component Layer* of the AHAM model.

[9]This initial version actually had no name or version number.
[10]Named AHA!, but still without a version number.

3.2.3.2 AHA! version 1.0

For AHA! version 1.0 the page format was extended with some XML tags, and some XML formats were introduced: for headers and footers, for the “requirements” (like prerequisites and inhibitors) and for “generate rules” (adaptation rules, defining updates to the “knowledge” value for concepts).

The above conditional fragment would become:

```
<if expr="definition and history">
<block>
  This part appears if the two "concepts" definition and history
  are both known according to the user model.
</block>
<block>
  If this is not the case then this alternative is presented instead.
</block>
</if>
```

AHA! would parse documents with these tags and generate output that was plain HTML.

Functionally the main differences between AHA! 1.0 and its predecessors were that the requirements and adaptation rules were separated from the content, and that the rules could update knowledge with a positive or negative value, thus no longer enforcing monotonic behavior.

From the AHAM viewpoint, the rules defining user model updates, and the requirements defining the link adaptation have been pulled out of the *Within-Component Layer* and moved to the *Storage Layer*. However, the content adaptation is still hidden in the pages.

3.2.3.3 AHA! version 2.0

A number of issues have been introduced in AHA! version 2.0 [De Bra et al., 2002a,b,c; Stash and De Bra, 2003]:

1. In AHA! 2.0 the use of XHTML was introduced. Standard XHTML was augmented with a module[11] to allow AHA!-specific tags. (They were used in AHA! 1.0 but only as an informal extension to HTML.) The new tags could be used for the generation of optional headers and footers and for conditional inclusion of fragments. The AHA! engine looked at the AHA! tags to perform the conditional inclusion, and produced a document in standard XHTML format. In order to use modularised DTDs[12] (and specifically the XHTML DTD extended with an AHA! module) the Xerces XML

[11] http://www.w3.org/TR/2000/CR-xhtml-modularization-20001020/

[12] Document Type Definition

parser[13] was used (instead of the SAX parser[14] used before, that could not handle them). With the modularized XHTML (and the Xerces parser) the processing overhead became larger because of the parsing time for the XHTML DTD, but this proved not to be a problem (thanks to hardware becoming faster as well).

In order to support a rich user model with arbitrary attributes for concepts the syntax of expressions in the <if> statements was changed. It had to take into account values of particular attributes, for example:

```
<if expr="c2L690_definition_knowledge > &&
          c2L690_history_knowledge > 50">
```

The names of the concepts have a prefix which is the name of the application they belong to, and a suffix which is the attribute to be used. Expressions in AHA! 2.0 (expressions in the <if> statements in the pages, requirements for concepts and requirements for actions in the concepts structures) were evaluated using the open source JEP parser[15] (Java Mathematical Expression Parser). JEP is a Java API for parsing and evaluating mathematical expressions. With this library an arbitrary formula entered as a string can be evaluated. JEP supports user defined variables, constants, and functions. A number of common mathematical functions and constants is included. AHA! was forced to introduce a restriction that the name of the application, and the concept names (after prefix) should start with a letter. Otherwise the parser will not be able to evaluate expressions (thinking that every string that starts with a number is a number). Because of this restriction the letter "c" was added in the previous expression for online course 2L690 provided by the TU/e.

2. AHA! 2.0 also introduced the ability to adapt HTML pages from other servers. The reason to allow external HTML documents in an adaptive application is that applications like on-line courses may wish to not just refer to but also adapt pages from other sources. However, these external pages should be valid HTML, because AHA! 2.0 parsed them with the javax.swing.text.html.parser that would produce error messages for pages with (common) HTML errors.

Despite the many new features and added adaptation and user modeling capabilities of AHA! 2.0, the separation between the *Storage Layer* and the *Within-Component Layer* was still exactly the same as in AHA! 1.0.

3.2.3.4 AHA! version 3.0

A number of new features have been introduced in AHA! version 3.0 [De Bra et al., 2003a, 2004b,c, 2006]:

[13] http://xerces.apache.org/xerces-j/
[14] http://java.sun.com/j2se/1.5.0/docs/api/javax/xml/parsers/SAXParser.html
[15] http://www.singularsys.com/jep/

1. In AHA! 3.0 an alternative way to perform the conditional inclusion of fragments was introduced, by using the <object> tag. (AHA! can still work with the old format supporting <if> tags.) The object tag exists in XHTML to (unconditionally) include an object stored in a separate file, so no XHTML extension was needed to allow this tag to be used for the *conditional* inclusion of objects.

   Fragments can be turned into separate resources linked to particular objects and represent these objects under various conditions. Each resource should be a well-formed (XHTML) document[16] in order to be parsed and scanned for the inclusion of more objects. This new way of handling fragments has an important advantage over inserting the fragment text into the page: when a fragment is used many times in one adaptive application, it only needs to be defined/maintained in one place.

   The AHA! engine treats object tags with a special "aha/text" (or "text/aha") type as conditional:

   ```
   <object name="ConditionalObject" type="aha/text" />
   ```

   The engine uses rules that are defined in the concept structure to decide which actual base-object to include. As a result conditional fragment inclusion is realized without the need for extensions to the XHTML language. Furthermore, since nothing in AHA! still depends on the XHTML language, other XML formats can be parsed and served as well. The only requirement for AHA! to perform content adaptation is the availability of an <object> tag, or another tag that can be used for the same purpose (and that must be defined as such in the AHA! software). For example, in SMIL we can use the <ref> tag for the conditional inclusion of objects (because SMIL does not have the <object> tag but uses the <ref> tag for a the same purpose, with the same syntax):

   ```
   <ref src="ConditionalObject" type="aha/text" />
   ```

   This results in realizing the adaptive multimedia presentation and adaptation of modality techniques as specified in the Brusilovsky taxonomy [Brusilovsky, 2001].

   Apart from HTML and XHTML, link adaptation can be performed to documents in other XML formats that have an anchor tag (called <a>) and that allow for the inclusion of stylesheets.

2. The JEP parser used in AHA! 2.0 for evaluating expressions has an important limitation: it requires a static variable table in which all the values are stored. In AHA! 3.0 we wished to introduce non-persistent attributes, of which the "value" is

[16] A document is well-formed when it is structured according to the rules defined in Section 2.1 of the XML 1.0 Recommendation (http://www.w3.org/TR/REC-xml/#sec-well-formed). Essentially this means that all elements must have closing tags and nest properly.

an expression that must be evaluated whenever the attribute value is needed. A new parser was created to allow variables that are actually expressions (using the default values of attributes). An example where this is used is the "suitability" attribute of concepts. As this is a non-persistent attribute and therefore is not stored in the user model its default will be taken. The default can be just trivial — true or false. But it can also be an expression using the attribute values of other concepts. For example, as has been shown previously, in the "tutorial" application the default of the suitability attribute for the concept "welcome" is "tutorial.readme.knowledge>50". This expression can be evaluated on the fly, something the old JEP parser could not do.

Regarding the separation of concerns the main difference between AHA! version 2.0 and 3.0 is the use of the <object> tag, and the related processing as part of the adaptation model, has completed the separation between the *Storage Layer* and the *Within-Component Layer*. All the aspects related to user modeling and adaptation can be specified completely outside the *Within Component Layer* (although for convenience the old <if> tags for the conditional inclusion of fragments can still be used). The step towards enabling AHA! to process arbitrary XML formats (and not just XHTML and SMIL) has become a very small step now.

3.2.4 Summary of AHA! in relation to the AHAM model

In this section we provided a detailed overview of the AHA! components, such as domain, adaptation and user models and within-components layer, in relation to the AHAM model.

We discussed that, in contrast with the AHAM model, the domain and adaptation models of AHA! are combined. However this is the case only at the low level, used by the AHA! engine. At the higher level, used by the Graph Author, these models are separated. The tool allows to separate the definition of concepts (see 3.2.1.1) and concept relationships (see section 3.2.1.2) comprising the domain model and adaptation rules (see 3.2.1.3) comprising the adaptation model. Keeping these definitions separate at the conceptual level the Graph Author also translates them into the format that the AHA! engine can understand with DM and AM combined.

In section 3.2.1.1 we have seen that AHA! 3.0 added a new type of concepts that corresponds to AHAM fragments — object concepts. As discussed in 3.2.3 conditional objects being included into the content pages using the standard XHTML "object" tag were added as an alternative to within-page fragments being included using AHA! "if" tags. The inclusion of objects is more complex than the inclusion of fragments and thus is more complex than what was envisaged in the AHAM model. In AHAM selecting fragments has no effect on UM. In AHA! the other hand, access to an object concept, as to any other concept in AHA!, generates UM updates. Therefore AHA! conditional objects are really an extension to the AHAM model. Moreover, the new way of handling fragments on the pages led to the separation between the Storage Layer and the Within-Component Layer of AHA! as proposed by AHAM.

Furthermore, through the section we addressed the design requirements “aspects used for adaptation”, “arbitrary concept relationships”, “generic adaptation rules”, “non-monotonic user model updates and cycles in adaptation rules” formulated in section 2.8. AHA! answers all the design requirements except for layout adaptation which is not possible in AHA! yet.

3.3 AHA! Authoring Tools

As discussed in section 2.8 of chapter 2 it is desirable for a general-purpose tool to provide a generic authoring environment offering a user-friendly interface for designing various types of applications. In this section we present two Java Applet based authoring interfaces provided by AHA! — the Graph Author and the Concept Editor — and discuss their positive sides and limitations. We explain in which cases which tool is more advisable to use. The Concept Editor is a tool that follows the design of the AHA! engine: it is used to define concept structures and to associate event-condition-action rules to (attributes of) concepts in order to define the adaptation the engine needs to perform. The Graph Author follows the AHAM structures: it is used to design a concept hierarchy and a graph of concept relationships. The required adaptation behavior follows from a translation of the concept relationships to the low level rules, through templates that define generic adaptation rules. So these tools represent an “assembly language” versus a “high level programming language” approach.

Both Graph Author and Concept Editor produce a server side XML file describing the conceptual structure of the application. Figures 3.9 and 3.11 provide examples of this structure in pseudo-code. The actual syntax of the resulting file is quite verbose and may be hard to read, though a technically skilled author can manually edit such files. Authoring tools allow to hide the XML syntax of the AHA! adaptation language from the author, thereby making it unnecessary to study this syntax in order to be able to create an adaptive application.

Below we show some examples of entering the conceptual structure using both tools. We use the same “tutorial” example as in figure 3.6.

3.3.1 The Graph Author

The Graph Author tool was proposed in [De Bra et al., 2002]. It allows the author to create the domain/adaptation model as a (labeled) graph of concept relationships.

Figure 3.13 shows the interface of the Graph Author with the graph representing the structure of prerequisite relationships in our “tutorial” application. Other types of relationships are filtered out to prevent clutter.

The Graph Author window is split into two parts, showing the concepts (as a hierarchy) on the left and showing the concept relationships (as a graph) on the right. The concepts are structured as a hierarchy which in fact also is a structure of concept relationships (and always present in an AHA! application).

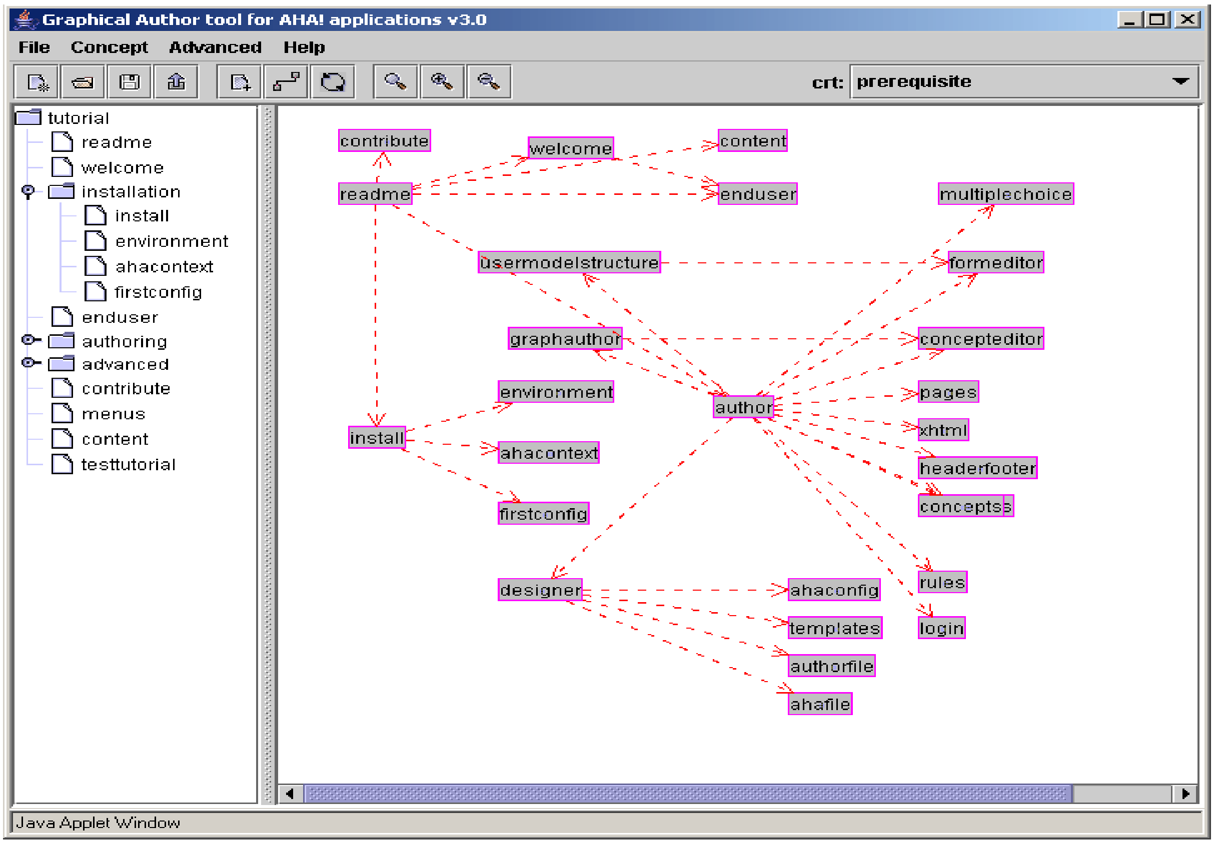


Figure 3.13: Screen shot of the Graph Author

The author can select and apply the relationships from a list of predefined types, such as prerequisite or inhibitor relationships. A technically skilled author can define new types of relationships (s)he might need for his/her applications by creating the corresponding templates as was discussed in 3.2.1.2. The templates are defined using an XML notation; there is no authoring tool for this yet. This involves defining the translation of concept relationships to AHA! adaptation rules, which is much more a matter of understanding how the adaptation and user model update process works than of syntax. Since the templates define *generic adaptation rules*, the Graph Author is called a *high-level* authoring tool.

The Graph Author knows how to combine different relationships between the same concepts into correct AHA! low-level adaptation rules that express the meaning of all the given rules. The Graph Author can show the graph for each concept relationship type separately or can overlay the graphs using different colors or line styles.

Figure 3.14 shows a dialog box for creating/editing a concept in the Graph Author.

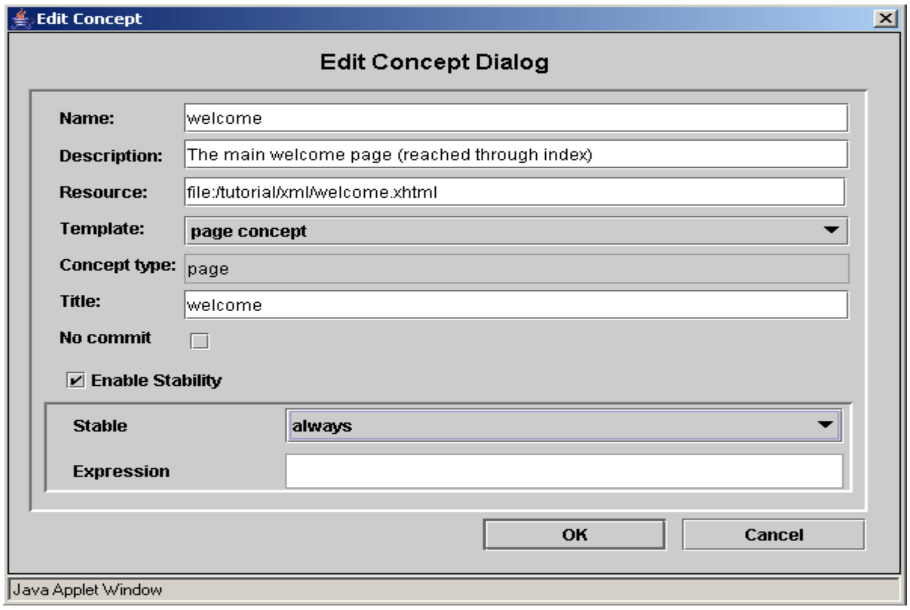


Figure 3.14: Dialog box for creating/editing a concept

The author has to specify the name, description and title for a concept, select its template that will automatically define the concept type, attributes and whether the author should specify a resource. There are also the "no commit" and "stability" options. "No commit" means that user model updates associated with the access to the concept are not committed to the user model database. Other applications that inspect the user model can thus not see these updates. By default this option is not enabled. Stability is used to keep the conditionally included objects the same, even though user model updates might suggest to include different objects later.

When a concept is defined using a template that has a showability attribute (or when a showability attribute is added manually) the resource can be selected based on expressions. Figure 3.15 shows the resource dialog box.

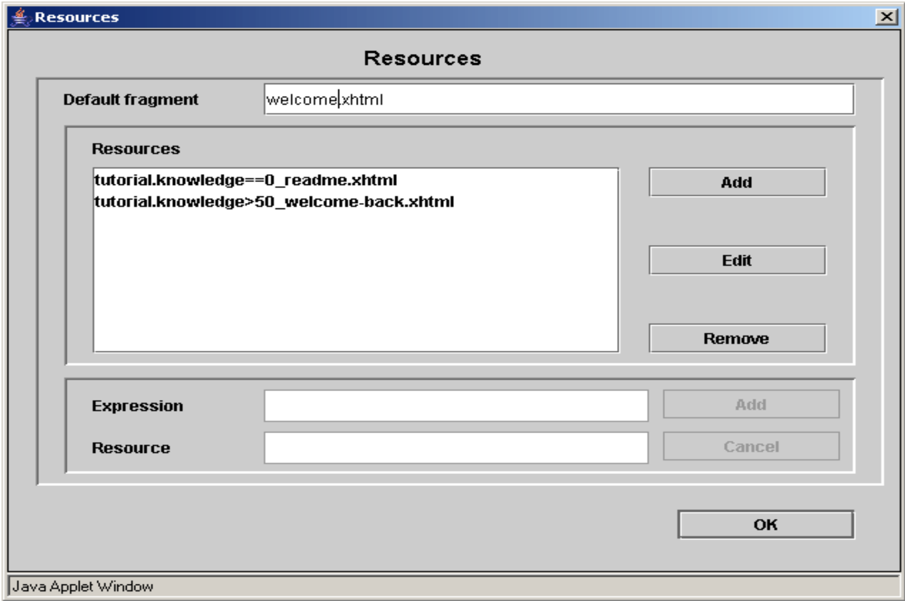


Figure 3.15: Dialog box for performing resource selection

When a concept is accessed the resource to be presented to the user is determined by examining Boolean expressions (from top to bottom as entered in figure 3.15). If none of the conditions are satisfied then a default resource is presented. This construct can be used for deciding which page to present when following a link to a concept, but also to decide which resource to include when the <object> tag is used to conditionally include an object.

The domain/adaptation model created through the Graph Author closely resembles the definition of a domain and adaptation model in AHAM. The Graph Author generates two files. One file describes the structure of the domain model including the high-level relationships between the concepts. The other file contains the low-level adaptation rules explained in the next section. The AHA! engine operates at this lower level.

Sometimes an author may wish to perform authoring at a still higher level, for example, adding a certain type of adaptive behavior to the whole domain without the need to draw the graph of the same types of relationships between all the domain model concepts. It makes sense to realize this by creating extensions of the Graph Author so that higher level constructs and graph-author-level constructs can be combined. In chapter 6 we show how we extended the Graph Author to allow the application of adaptation strategies corresponding to various learning styles.

3.3.2 The Concept Editor

The "Concept Editor" or "Generate List Editor" is a *low-level* authoring tool. It allows to control every bit of functionality of AHA! (regarding concepts, attributes and adaptation rules). Using the Concept Editor the author has to write all the individual adaptation rules himself/herself. This corresponds to the creation of *specific adaptation rules* in the AHAM model rather than *generic* ones defined through the Graph Author templates.

Figure 3.16 shows the interface of the Concept Editor.

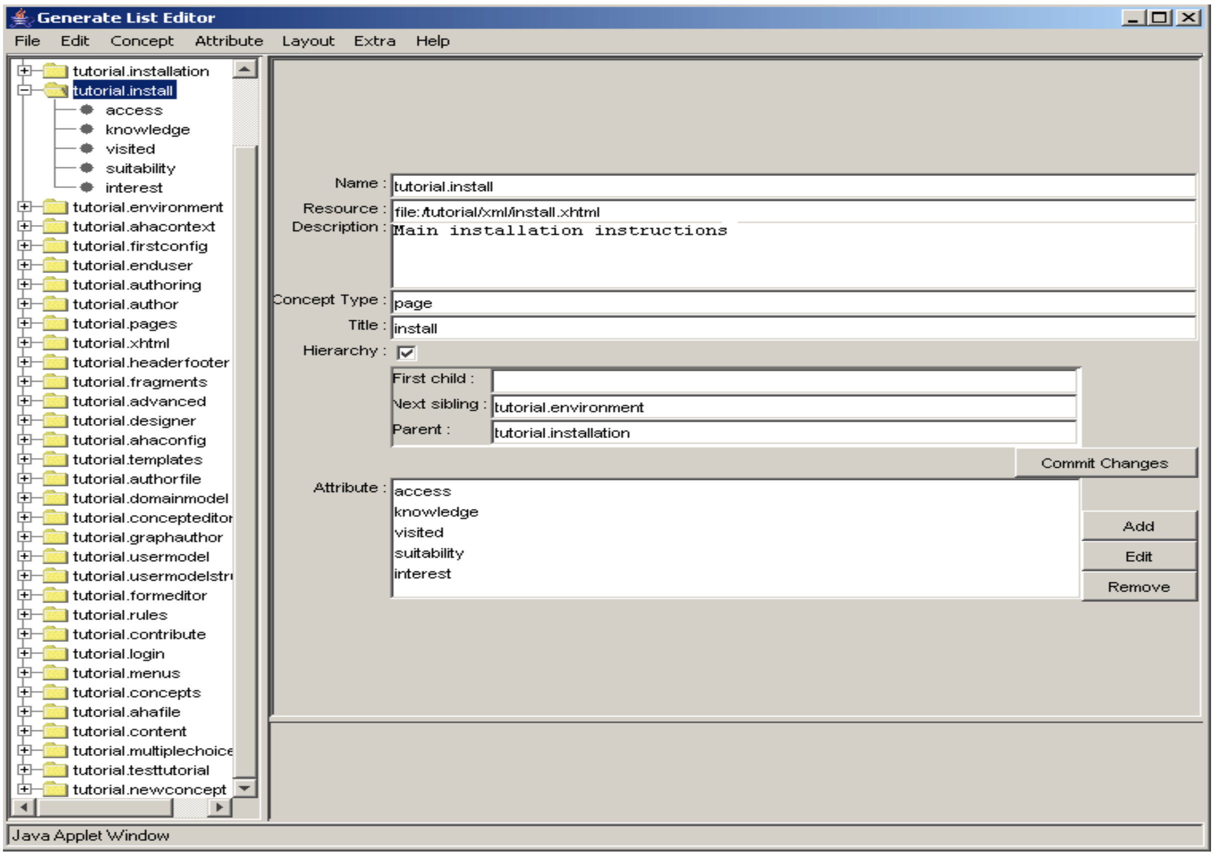


Figure 3.16: The Concept Editor

The concept hierarchy is represented inside the concepts, but the Concept Editor (unfortunately, due to the implementation limitations) does not show that hierarchy in its left frame. The editor lists all the concepts (of a single application) on the left, and shows details of a selected concept ("tutorial.install" in the example) on the right.

The author has to specify the name, description, title, concept type, possibly resource and attributes of the concept in addition to the ones which were added using the concept template. Additionally the author has to select whether a concept should be a part of the hierarchy (and therefore shown in the table of contents, used for knowledge propagation, etc.) or not. If a concept is defined as part of the hierarchy the author will have to specify its parent unless it is the top of the hierarchy, and possibly its children and siblings. The siblings should be arranged sequentially. (This order is used in an automatically generated table of contents.)

In figure 3.17 we look at the "access" attribute of the selected concept.

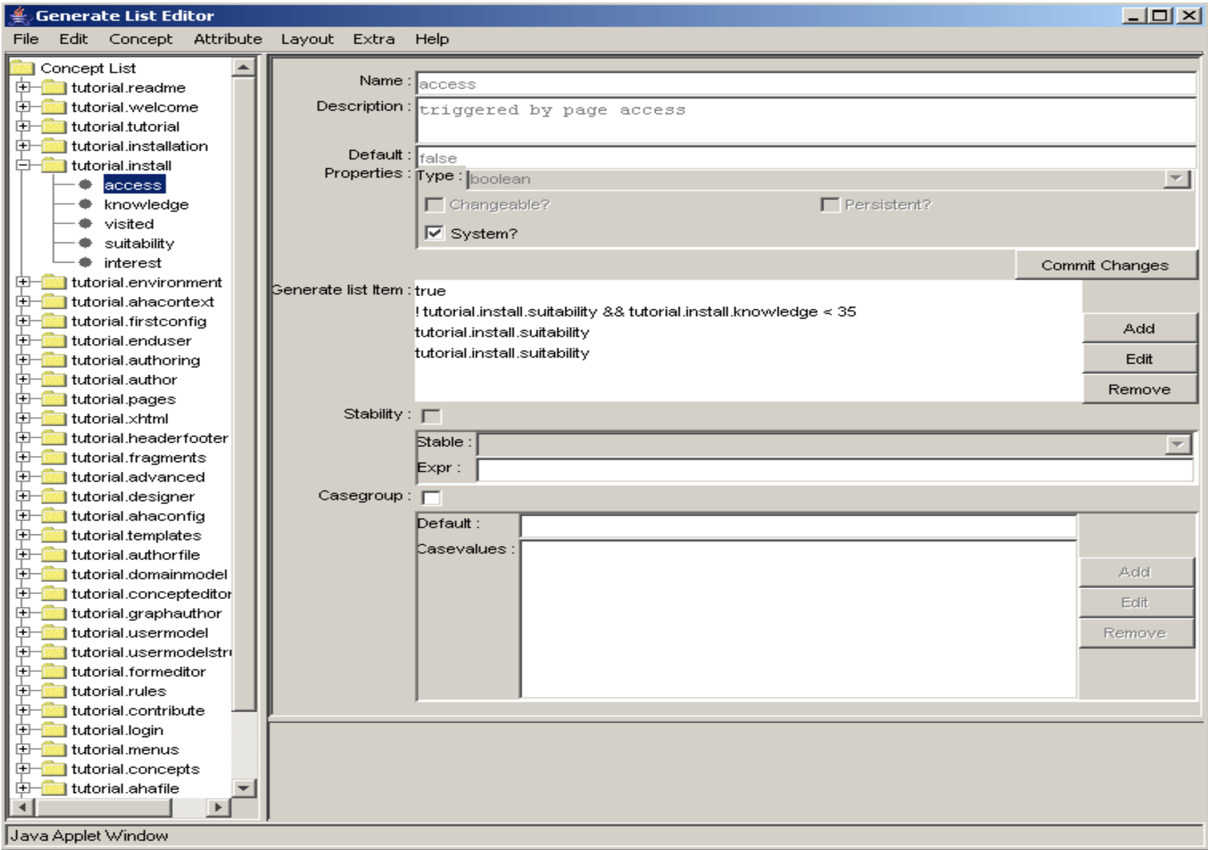


Figure 3.17: Information about the "access" attribute in the Concept Editor

As can be seen from the figure, "access" is a system-defined, non-changeable and non-persistent attribute. (Hence the template has determined the name, description and default and in the editor this cannot be changed without first making this attribute no longer a system attribute.)

Figure 3.17 shows the conditions of all the generate rules associated with the attribute ("Generate list item", one per line). When the author clicks on the condition and then "edit" (s)he gets to see the actions that are performed when the condition is true, and the optional actions for when the condition is false.

The author can choose the stability option and, in the casegroup, define the resources representing the concept.

Figure 3.18 shows the rule for the selected condition

```
!tutorial.install.suitability && tutorial.install.knowledge < 35
```

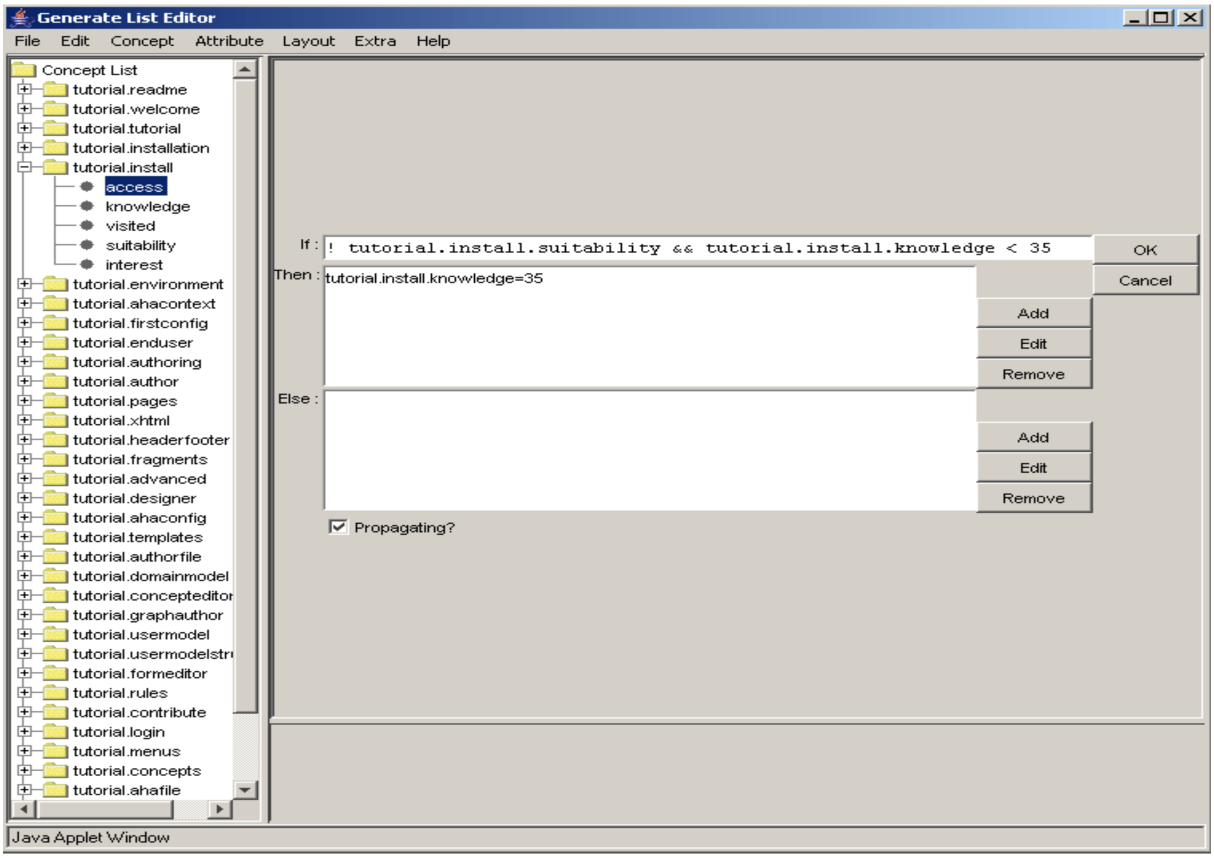


Figure 3.18: Editing an event-condition-action rule in the Concept Editor

The condition is shown in the "If" field. The set of actions to be performed if the condition is true is specified in the "Then" area. In the given example the knowledge of the "tutorial.install" concept is set to 35. The set of actions to be performed if the condition is false are specified in the "Else" area (no actions in the example).

For each action there is a "Propagating" attribute that indicates whether this action is allowed to trigger other adaptation rules or not.

Note that the Concept Editor can read the files containing the low-level adaptation rules produced by the Graph Author, but not vice versa. It is impossible to deduce from a set of adaptation rules the set of high-level relationships that lead to these rules.

3.3.3 Summary of AHA! authoring tools

In this section we presented two authoring tools provided by AHA! — the high-level Graph Author and the low-level Concept Editor. Both of them can be used for creating various types of applications. In both tools, the author's task is made easier by predefining a common structure for concepts. When the author adds a concept a default set of attributes and rules is created. This not only reduces the amount of work for the author, but it also helps to avoid common (beginners) mistakes.

The Graph Author is a more recent tool than the Concept Editor. It has a much more user-friendly interface which makes it more suitable for novice authors who only want to use the predefined relationship types in their applications. The Graph Author is also very convenient for technically skilled authors having XML knowledge who want to create and apply new types of relationships in their applications. On the other hand, the Graph Author does not let the author see the low-level adaptation details. These can be only seen either in the generated XML file containing the domain/model structure of the application or when opening the application in the Concept Editor. However, if the author makes some changes to the application structure in the XML file or in the Concept Editor, then opens it again in the Graph Author and saves the application, all the previous updates will be lost. They will be overwritten by the adaptation rules specified in the Graph Author templates.

If the author wants to be able to control every bit of the application behavior then (s)he may be advised to use the Concept Editor. In that case the author will be able to control the rules execution order, rules propagation, etc.. A drawback of using the Concept Editor for defining rules is that frequently occurring concept relationships lead to a lot of repetitive work (in that case the author is advised to use the Graph Author). Also, these relationships are described through their effect on the user model only, and are thus not visible as "relationships". An author who is revisiting an application through the Concept Editor may not recognize which relationships are expressed by which adaptation rules and may thus not recognize the higher-level structure that is hidden in the large set of rules. Sometimes it can also be quite difficult to describe the intended user model and adaptation through the adaptation rules.

A combination of the features of both tools would be nice. It would be good to extend the Graph Author with the ability of having access to the low-level adaptation details and being able to save the updates to them.

3.4 AHA! Performance

The performance of adaptive hypermedia systems is not often discussed. We have mostly anecdotal information from watching demonstrations and playing with other research systems. Suffice it to say that all systems (except for AHA!) show a significant delay between a user action, such as clicking on a link, and the appearance of the result on the screen. Users of the Web are known to be impatient. We have not evaluated how impatient users really are but instead concentrated on providing an experience that is virtually indistinguishable from accessing a website with static pages. (We interpret this as meaning sub-second response time.)

AHA! is implemented entirely in Java, and works with the Java-based webserver Tomcat and with Java servlets. It runs on any full implementation of the Java 2 platform. AHA! is frequently tested on Microsoft Windows (2000 and XP) and Linux. Linux is the preferred platform, not just because it is Open Source, as is AHA! but also because it is better at multitasking. This is especially noticeable on dual cpu or dual core machines. AHA! can be used on slow computers with little memory. (We have tested it on a Pentium 166 with 32 MB ram running Windows'98.) However, for an enjoyable experience a faster server is recommended. On a laptop with Pentium M processor and Windows XP we are currently achieving sub-second response times with the AHA! tutorial, which is an application using three frames (a "toolbox" frame, a frame showing part of the concept hierarchy and a frame showing the adapted page). Every click on a link requires the execution of 4 servlets (one for the frameset and three for the three frames), one of which triggers the user model updates and two of which require adaptation to be performed. The performance of a modest server running Linux can be judged by using the AHA! tutorial on the AHA! website[17] and is significantly better still.

The aim of AHA! is to take a minimum of time for processing the adaptation rules and generating the pages. Although very powerful, the AHA! rule system is very straightforward and can be implemented in an efficient way. Several optimizations were added, including caching of the concept structure and adaptation rules of the application (instead of reading their definition upon each request) and caching of the user model (instead of retrieving it from a database upon each request). Generated pages cannot be cached as they are adapted (possibly) differently upon each request. A minor drawback of these optimizations is that when an author changes the concept structure (and saves it from the Graph Author or Concept Editor) users who are already logged in do not see the changes until they log out. Also, if different applications share and update the same user model AHA! does not notice the updates performed by the other application (and may inadvertedly even overwrite them).

There are several issues that have to be taken into account while working with AHA!

1. With most browsers a page is not requested from the server when the user navigates through the browsing history (using the back and forward buttons). This means that the user will not see the result of adaptation. AHA! 3.0 generates HTTP response headers that tell browsers to revalidate (i. e., reload) pages at all times, but not all browsers honor this request.

2. The user should not try to work simultaneously with several applications opened in different windows. When the user logs in to the application the value of the attribute "course" of the concept "personal" is initialized. This value indicates the name of the application ("course") the user works with. In case the user starts several applications the value of this attribute will be the name of the last application (s)he logged in to. Only this last opened application will work correctly. Previously opened applications will use the wrong value of the attribute "course" from the user model. This will lead to unpredictable results making the applications not work properly. Applications do have access to the concepts (and attributes) of other applications running on the same AHA! server instance. So one on-line course can, for instance, perform adaptation based on knowledge the user gained in another course. It's just that the user cannot be logged in on both courses at the same time.

[17]http://aha.win.tue.nl/

3.5 Summary

The validation of AHAM using AHA! 1.0 described in [Wu, 2002] was quite positive in the sense of being able to represent the behavior of AHA! in the AHAM model. AHA! 2.0 brought AHA! closer to the AHAM model in terms of flexibility of the user model structures and adaptation rules, but not in terms of separation of concerns (the separation between the *Storage Layer* and *Within-Component Layer*. AHA! 3.0 has brought AHA! much closer to the AHAM model by removing more of the adaptation functionality from the *Within-Component Layer* and moving it to the *Storage Layer*.

Let us check the list of the design requirements for creating a general-purpose tool described in the previous chapter and see if they were realized in AHA! version 3.0.

1. *Adaptation types and techniques.* In 3.1.3 we showed that AHA! provides content and link adaptation, but no layout adaptation yet. It provides the majority of the adaptation techniques as specified in Brusilovsky's taxonomy [Brusilovsky, 2001]. The ones not implemented yet are natural language adaptation, sorting fragments, adaptive link sorting and adaptive link generation techniques. Furthermore stretchtext technique cannot be directly realized in AHA! but can be obtained by using AJAX and direct guidance is possible but tricky. On the other hand, AHA! 3.0 adds a novel technique called "adaptive link destinations" that is not specified in Brusilovsky's taxonomy.

2. *Aspects used for adaptation.* AHA! supports a rich user model with multiple concepts and arbitrary attributes of these concepts. Through these concepts and attributes it is possible to represent various types of information and use them for adaptation purposes (see section 3.2.2). However they can be used only for content and link adaptation but not for adaptation of the layout, as would be needed when adapting to different devices (with very different screen sizes).

3. *Arbitrarily concept relationships.* Through the Graph Author templates, authors can create various types of concept relationships they might need for their applications (see 3.2.1.2).

4. *Generic adaptation rules.* The Graph Author tool makes this possible through the use of templates (see 3.2.1.2).

5. *Handling non-monotonic user model updates and cycles in adaptation rules.* Handling non-monotonic user model updates became possible in AHA! 1.0, but was generalized to the event-condition-action rules in AHA! 2.0, and left unchanged in version 3.0. As of AHA! 3.0, the system provides three options for handling cycles and non-confluence in adaptation rules — by making the order of rules execution visible in the Concept Editor tool, by an automatic check for cycles in the authoring tools and by limiting the number of objects included in the page (see 3.2.1.3).

Furthermore AHA! addresses other issues, such as:

- *reasonable performance.* In section 3.4 we discussed how AHA! achieves a sub-second response time — a delay between a user action, such as clicking on a link, and the appearance of the result on the screen. It thus provides an experience undistinguishable from accessing a website with static pages.

- *authoring environment for designing various types of applications.* AHA! provides two authoring tools — the high-level Graph Author and the low-level Concept Editor (see section 3.3). Through the templates supported by the Graph Author the authors can specify various types of concept relationships and generic or specific adaptation rules they might need in their applications. The Graph Author has a more user-friendly interface (than the Concept Editor) and allows to create the domain/adaptation model as a (labeled) graph of concept relationships. The Concept Editor also allows for specifying various types of relationships that might be necessary in the adaptive applications; however, it only supports the creation of specific adaptation rules. This allows to control every bit of functionality but on the other hand requires a lot of repetitive work while defining frequently occurring relationships.

Creating a general-purpose tool is a time consuming and difficult process. As can be seen not all of the required aspects are currently provided by AHA!. Nevertheless it can already be used in various application domains where content and link adaptation are required. Apart from tutorials and courses AHA! has been used to create adaptive talks (adaptive Powerpoint-like presentations), adaptive information kiosks (like in one version of the IShype information for interns and master thesis students) and even a research paper [De Bra et al., 2006].

Therefore an answer to research question 1 “Is an AHS necessarily a special-purpose tool?” is “No”. We have shown this by implementing the AHA! system. We have presented how an educational AHS AHA! 1.0 has been turned into a powerful tool that supports the creation of applications targeting various domains and not necessarily educational settings.

Rich user models and the possibility to define generic adaptation rules made it possible to think of new extensions to the AHA! system, such as support for learning styles. However, the “standard” general-purpose authoring tools offered by AHA! are not ideal for creating this support. In the following chapters, we introduce the field of learning styles, discuss their possible connections with adaptive hypermedia and show how we implemented them in AHA!.

Likewise, educational applications may require more than adaptation based on the “access” to concepts or pages. Other applications may interact with AHA!’s user model to offer adaptation based on that model, and to even add or change information to/in that model. That does not yet lead to a seemless virtual integration however. AHA! contains authoring tools and an engine for creating and executing (adaptive) multiple-choice tests (a tool developed at the University of Cordobà). We do not describe that in this thesis, but mention it because it shows that in order to make a general-purpose tool like AHA! really suitable for a specific application area some extensions may be necessary.

Chapter 4

Learning Styles (LS) Theory

In this chapter we introduce the field of learning styles (henceforth referred to as LS). We focus on those LS that (according to psychological research) can potentially apply to adaptive Web-based settings.

Section 4.1 discusses the origins of the learning styles theory. Section 4.2 provides definitions of the terms *cognitive* and *learning styles*. Section 4.3 reviews a number of learning style models that involve LS having potential implications on adaptation purposes and therefore being of interest for our research. Section 4.4 discusses the problems connected with assessing the learning styles of users. Section 4.5 presents some conflicting ideas concerning the impact of learning styles in education. Section 4.6 summarizes the chapter.

4.1 Introduction

What are the sources of the idea about the existence of learning styles? Do they really exist or this is only an illusion? What is their nature?

Looking at people around us we notice that not everyone sees the world in the same way we do. Different people can have a different view upon the same situation (the way how they perceive and estimate it) and correspondingly their response can be different (the way they make decisions and judgements). Despite these differences everyone can be right in his/her own way. It would be possible to say that each person is clever in his/her own manner.

If we look at the educational area we see that students in schools and at university vary enormously in the speed and manner of picking up new information and ideas, and in applying knowledge under new circumstances. Students learn in many different ways. For instance, look at the way in which particular learners prefer to study a foreign language. Some have a preference for hearing the language (the so-called *auditory* learners), some for seeing it written down (*visual* learners), some for learning it in discrete bits (*analytic* learners), some for experiencing it in large chunks (*global* or *holistic* or *experiential* learners), and many prefer to do something physical whilst experiencing the language (*kinesthetic* learners).

Another example — the ways how different students learn a new programming language. Some students may start writing the programming code immediately after learning the very basics of the language (*active* learners), whilst others may need to read the tutorial and look at a number of examples before writing any code (*reflective* learners).

There are probably as many ways to teach as there are to learn. Some teachers only lecture, others involve students in discussions, some concentrate on rules, others give more examples, some focus on memorizing of the material, others on understanding [Felder and Henriques, 1995].

We see these individual differences in many aspects of everyday life. But what are the underlying psychological mechanisms of the individual mentality? Cognitive psychology deals with this question.

The studies in the area of cognitive/learning styles began in the mid 20th century. The term "cognitive styles" appeared first in the works of American psychologists referring to the individual peculiarities of perception, analysis and categorization [Witkin et al., 1971; Gardner et al., 1959; Kagan, 1966]. In the 80-90's the concept of cognitive styles was extended with the advent of new styles, such as "thinking styles" [Grigorenko and Sternberg, 1995], "learning styles" [Kolb, 1984; Honey and Mumford, 1986] and "epistemological styles" [Wardell and Royce, 1978]. Moreover, at that time some theorists proposed the so-called "meta-styles" to supersede the whole multitude of existing cognitive/learning styles, e. g., verbalizer-imager and wholist-analytic styles [Riding, 1997] (see 4.3).

The theory and practice of CS/LS has generated great interest, however much controversy as well. The confusion starts from the fact that there is no single definition for the term cognitive/learning style. Researchers in the field of learning styles tend to interpret theories in their own terms and there is no "one" commonly accepted theory and vocabulary of terms. The terminology varies from author to author, though by different terms they can actually mean the same characteristic or concept. Furthermore there is conflicting evidence concerning the degree of stability of learning styles, possibility of their reliable and valid measurement and their impact on education or learning.

Throughout the history of research on cognitive/learning styles new studies, instead of answering existing questions, often brought even more contradictory facts and added more confusion. Many researchers who started their research with much enthusiasm felt disappointment. Some rejected the style construct as an illusion, or at best as a construct which is impossible to operate with and therefore undeserving of further research [Freedman and Stumpf, 1980]. Until now the psychology of cognitive styles still remains a weakly developed research field, being rather at the stage of its formation [Holodnaya, 2002].

Despite their complex nature the theory of CS/LS attracted much attention among the researchers in the area of adaptive hypermedia, including the authors of this dissertation. It provides us with interesting ideas for further direction of our own research. Furthermore the approaches we are going to work on can be later used by psychologists to test their theories about application of LS in education.

In the following section we look at existing definitions for cognitive and learning styles provided by different theorists.

4.2 Definitions of Cognitive/Learning Styles

4.2.1 Cognitive styles

In the cognitive psychology literature we found the following definitions for what the *cognitive styles)* are:

- consistent and enduring differences in individual cognitive organization and functioning [Ausubel et al., 1978];
- an expression of psychological differentiation within characteristic modes of information processing [Witkin and Goodenough, 1981];
- an individual's characteristic and consistent approach to organizing and processing information [Tennant, 1998];
- the way the individual person thinks and an individual's preferred and habitual approach to organizing and representing information [Riding and Rayner, 1998].

On the basis of these existing definitions in our point of view a cognitive style is *an individual's consistent approach in perceiving, remembering, processing, organizing information and problem solving.*

4.2.2 Learning styles

Learning styles are described by different researchers as:

- distinctive behaviors which serve as indicators of how a person learns from and adapts to his environment [Gregorc, 1979];
- preferences for one mode of adaptation over the others; but these preferences do not operate to the exclusion of other adaptive modes and will vary from time to time and situation to situation [Kolb, 1981];
- *composite of characteristic cognitive, affective and psychological factors that serve as relatively stable indicators of how a learner perceives, interacts with and responds to the learning environment* [Keefe, 1979].
- a description of the attitudes and behavior which determine an individual's preferred way of learning [Honey and Mumford, 1992];
- a coherent whole of learning activities that students usually employ, their learning orientation and their mental model of learning [Vermunt, 1996].

In most situations the terms cognitive and learning styles are used interchangeably by different theorists. Still there is a difference between them. Cognitive style deals with the "form" of cognitive activity (i. e., thinking, perceiving, remembering), not its content. Generally, cognitive styles are more related to theoretical or academic research, while learning styles are more related to practical applications [Liu and Ginther, 1999]. According to Riding and Cheema [Riding and Cheema, 1991] "learning styles are probably best regarded as an extension to cognitive styles to distinguish the act of learning from simple processing of information".

The definition for LS given by Keefe sees learning style as a broader construct, which includes cognitive along with affective and psychological styles. It was accepted by the leading theorists. We will therefore refer to the *definition by Keefe* in this dissertation.

Furthermore [Liu and Ginther, 1999] note that "a major difference between the terms cognitive and learning styles is the number of style elements involved. Specifically, cognitive styles are more related to a bipolar dimension while learning styles are not necessarily either/or extremes".

In this dissertation we will refer to the terms cognitive and learning styles interchangeably. Mostly we will use the broader term — learning styles.

It is necessary to make a distinction between styles and ability. According to Sternberg [Sternberg, 1999] an ability "refers to how well someone can do something". A style "refers to how someone likes to do something". A style therefore is "a preferred way of using the abilities one has". "We do not have a style, but rather a profile of styles".

There exist a wide range of various CS/LS. Different LS were defined within LS models developed by different psychologists. The complete review of LS models and LS they involve is beyond the scope of this dissertation. In the following section we discuss only those models that involve LS being of interest for our research.

4.3 Overview of LS Models

As a basis for our review we took the latest major report on learning styles theory provided by a team from Newcastle University, UK in 2004 [Coffield et al., 2004]. The report states that Mitchell [Mitchell, 1994] claimed there were over 100 learning style models. However Coffield et al. found only 71 of them worth consideration.

Different models involve various learning styles. A large number of models were produced by different groups of researchers working in relative isolation from each other. Therefore there is much conceptual overlap between them. Out of 71 identified models Coffield's report categorizes 13 as "major models using one or more of the following criteria:

- their theoretical importance in the field as a whole;
- their widespread use, either commercially or academically;
- their influence on other learning style models".

In the following we discuss 6 of these major models and 3 models presented in the report but not categorized as major ones.

Coffield's report organizes the LS models within the "families of learning styles". The grouping into certain families aims at capturing "the extent to which the authors of the model claim that styles are constitutionally based and relatively fixed, or believe that they are more flexible and open to change". This is not necessarily perfect and the only existing type of grouping. For instance, another very popular classification was provided by Curry [Curry, 1983]. We follow Coffield grouping as it is the most recent one, includes the latest developed LS models and tries to alleviate the disadvantages of previous classifications.

Figure 4.1 shows the families of LS as organized in Coffield's report and only those LS models within each family that we are going to discuss further on.

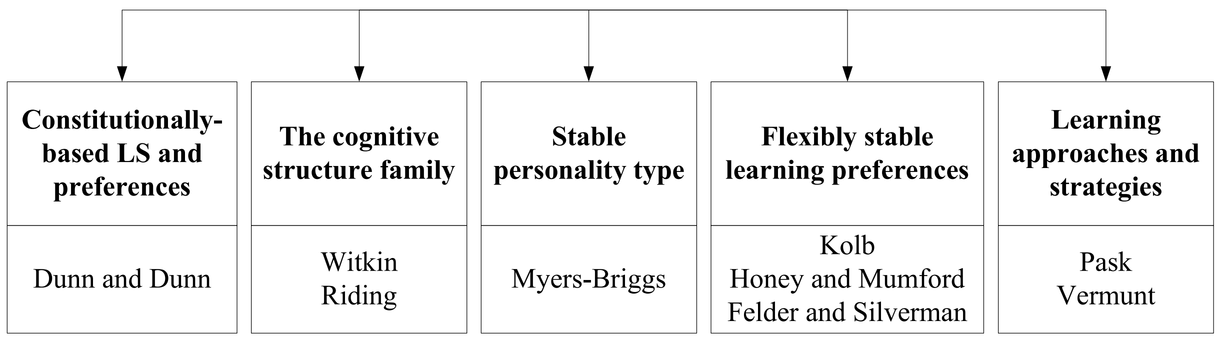


Figure 4.1: Learning styles families

4.3.1 Constitutionally-based learning styles and preferences

The Coffield's report includes in this family the models in which authors consider that cognitive/learning styles are fixed or at least are very difficult to change. "To defend these beliefs, theorists refer to genetically influenced personality traits, or to the dominance of particular sensory or perceptual channels, or to the dominance of certain functions linked with the left or right halves of the brain" [Coffield et al., 2004].

For example, Rita Dunn argues that learning style is a "biologically and developmentally imposed set of characteristics that make the same teaching method wonderful for some and terrible for others" [Dunn and Griggs, 1988].

Rita and Kenneth Dunn are the authors of the Dunn and Dunn learning styles model which became very popular internationally and is widely applied in the United States. It involves a very broad range of learning styles.

According to the Dunn and Dunn model, "learning style is divided into 5 major strands called stimuli. The stimulus strands are elements that significantly influence how many individuals learn" [Dunn, 2003a]:

- *Environmental* — this strand incorporates individuals' preferences for the elements of sound, light, temperature, and furniture or seating design.

- *Emotional* — focuses on students' levels of motivation, persistence, responsibility, and need for structure.

- *Sociological* — addresses students' preference for learning alone, in pairs, with peers, as part of a team, with either authoritative or collegial instructors, in a variety of ways or in routine patterns.

- *Physiological* — examines perceptual strengths (visual, auditory, kinesthetic or tactile, often abbreviated as VAKT), time-of-day energy levels, and the need for intake (food and drink) and mobility while learning.

- *Psychological* — incorporates the information-processing elements of global versus analytic and impulsive versus reflective behaviors, hemispheric elements.

For our research we choose only those elements that, according to psychological researchers, have applicability in the adaptive Web-based settings. The interesting elements are the sociological strand (possible applications will be discussed later), the sensory modality component from the physiological strand and the psychological strand.

The Dunn and Dunn model draws much attention to the field of modality preference. The sensory modality component is included in the physiological strand. According to [Bissell et al., 1971], a sensory modality is a system that interacts with the environment through one of the basic senses. The Dunn and Dunn model deals with the most important sensory modalities:

- *Auditory* — the students with the auditory style learn best through verbal lectures, discussions, listening what others say. Written information may have little meaning until it is heard.

- *Visual* — visual students learn through seeing. This modality is actually split into two, indicating preference to perceive materials as pictures or as text.

- *Tactile* and *Kinesthetic* — these two modalities are very closely related. The students of these types learn best through a hands-on approach: they underline and/or take notes. Particularly "tactile" refers to touch and "kinesthetic" to movement.

The physiological strand of Dunn and Dunn model deals with the following styles:

- *Analytic* — analytic students prefer to learn one detail at a time in a meaningful sequence. Once they know all the parts, they put the parts together and comprehend the "big picture".

- *Global* — global students are concerned with the whole meaning and the end results. They need to start with an overview of the "big picture" before they deal with elements of the whole.

  According to Rita Dunn [Dunn, 2003a]: "the majority of students at all academic levels are global rather than analytic, they respond better to information taught globally than they do to information taught analytically. ... Integrated processors can internalize new and difficult data either globally or analytically but retain it only when they are interested in what they are learning."

- *Impulsive* — impulsive students act on impulse and make decisions quickly without thinking too much.
- *Reflective* — reflective students cautiously reflect upon the implications of any proposed action, think about the various alternatives and evaluate each before making a decision.

The Dunn and Dunn model provides a number of instruments to measure these dimensions. The assessment identifies strong preferences, preferences, non-preferences, opposite preferences and strong opposite preferences. Each person's unique combination of preferences comprises his or her learning style.

The psychological strand of Dunn and Dunn model also includes *hemispheric elements*.

Research into brain functioning has quite a long history. It has revealed that the two hemispheres of the brain perform different functions. Much of this knowledge comes from the split-brain research carried out by Roger Sperry, winner of the Nobel Prize [Sperry, 1964]. His experiments revealed what Sperry described as "two spheres of consciousness" locked in the one head, the left-hand side having speech and a rational, intellectual style, while the right was inarticulate, but blessed with special spatial abilities" (quoted from [McCrone, 2000]).

According to Sonnier [Sonnier, 1991] hemispheric preferences might be a major contributing factor to individual differences. A popular view is provided by Springer and Deutsch [Springer and Deutsch, 1989] who consider that the *left hemisphere* is responsible for *verbal, linear, analytic thinking*, while the *right hemisphere* is more *visuospatial, holistic and emotive*.

Although hemispheric preferences are considered an important dimension of the Dunn and Dunn model, they do not provide any instrument for measuring hemispheric dominance.

Some other psychologists connect the behavior of their learning styles with the brain activity. Kolb [Kolb, 1999] claims that the dimensions of his model (discussed below) *concrete experience* and *abstract conceptualization* reflect *right-* and *left-brain thinking* respectively. Researchers say the same about intuition-perceiving/sensing-judging elements of Myers-Briggs Type Indicator (MBTI) [Hartman et al., 1997], *holist/serialist* learning [Entwistle, 1998], Witkin's dimensions field-dependence/field-independence [Holodnaya, 2002]. However some works [Tinajero et al., 1993] note the influence of right-brain hemisphere on the evidence of the field-independent style of processing information. "Riding thinks of his global-analytic dimension as being completely unrelated to hemisphere preference (unlike his visual-verbal dimension). This illustrates the confusion that can result from linking style labels with "brainedness" in the absence of empirical evidence [Coffield et al., 2004]).

However despite its long history the research has not revealed any satisfactory explanation for individual differences in the personal characteristics associated with right- and left-brain functioning. Researchers point out that the concept of cerebral preference has some limited practical and theoretical implications (quoted from [Liu and Ginther, 1999]). Therefore we will not consider this dimension in our further research.

4.3.2 The cognitive structure family

In this family a number of theorists is presented who, according to [Coffield et al., 2004], have a shared view (implicitly or explicitly expressed) of learning styles as "structural properties of the cognitive system itself" [Messick, 1984]. For this group, styles are not merely habits, with the changeability that this implies; rather, "styles are more like generalized habits of thought, not simply the tendency towards specific acts ... but rather the enduring structural basis for such behavior" [Messick, 1984] and as such, are not particularly susceptible to training. For this reason, many of these styles are very similar to measures of ability. For the theorists in this family, styles are linked to particular personality features, with the implication that cognitive styles are deeply embedded in personality structure. The learning styles in this family tend to be expressed as bipolar constructs (quoted from [Coffield et al., 2004]).

In the following we discuss two models from the cognitive structure family: Witkin's field-dependence versus field-independence dimension (FD versus FI) and Riding's model.

4.3.2.1 Witkin's dimension - field-dependence versus field-independence

Witkin is the most influential member of the cognitive structure group. The bipolar dimensions of field-dependence/field-independence have had considerable influence on the learning styles discipline, both in terms of the exploration of his own constructs and the reactions against it which have led to the development of other learning styles descriptors and instruments [Coffield et al., 2004]. This is one of the most widely studied dimensions of CS with the broadest application to problems of education [Messick, 1976; Witkin et al., 1977].

The exploration of the FD/FI construct, also known as the global-articulated continuum, began in the 1940s with Herman Witkin's research on human perception of the upright [Witkin et al., 1977]. Two tests were used for this purpose — the Rod and Frame Test and the Body Adjustment Test. For further research the Group Embedded Figures Test was developed. Here we outline the essence of the experiments.

(Quoted from [Coffield et al., 2004]): The Rod and Frame Test involves sitting the participant in a dark room. The participant can see a luminous rod in a luminous frame. The frame is tilted and the participant is asked to make the rod vertical. Some participants move the rod so that it is in alignment with the tilted frame; others succeed in making the rod vertical. The former participants take their cues from the environment (the surrounding field) and are described as "field-dependent"; the latter are uninfluenced by the surrounding field (the frame) and are described as "field-independent".

The Body Adjustment Test is similar to the Rod and Frame Test in that it also involves space orientation. The participant is seated in a tilted room and asked to sit upright.

Again, field-dependent participants sit in alignment with the room, while field-independent participants sit upright, independent of the angle of the room.

The Group Embedded Figures Test is a paper and pencil test. The participant is shown a geometric shape and is then shown a complex shape which contains the original shape "hidden" somewhere. The field-independent person can quickly find the original shape because they are not influenced by the surrounding shapes; the opposite is true for the field-dependent person. The authors claim that results from the three tests are highly correlated with each other [Witkin and Goodenough, 1981].

The construct of FD/FI measured in the tests broadened to include perceptual and intellectual problem solving. It is used to describe how much a learner's comprehension of information is affected by the surrounding perceptual or contextual field [Witkin et al., 1977].

Researchers draw several conclusions about the strategies and approaches taken by FD and FI individuals:

- *field-independent* individuals are highly analytic, sample more cues inherent in the field and are able to extract the relevant cues necessary to complete a task. They tend to discern figures as discrete from their background, to focus on details, and to be more serialistic in their learning. They operate within an internal frame of reference and thrive in situations where they need to actively structure their own learning [Witkin et al., 1977].

  Field-independent learners are more individualistic and rule-oriented and less likely to seek peer input [Jones, 1993]. Field-independent learners are also more efficient information processors with better short term memory encoding, better long term recall, and more accurate performance on visual search tasks than field-dependent individuals [Davis, 1991].

- *field-dependent* individuals process information globally and attend to the most salient cues regardless of their relevance. Field-dependent individuals typically see the global picture, ignore the details, and approach a task more holistically. They tend to see patterns as a whole and have difficulty separating out specific aspects of a situation or pattern. Field-dependent individuals take a passive approach, are less discriminating, and attend to the most salient cues regardless of their relevance. They also operate within an external frame of reference and prefer situations in which structure and analysis is provided for them [Witkin et al., 1977].

  In educational situations, field-dependent learners' tendency to be influenced by their peers is critical as they prefer feedback and social sources of information [Jones, 1993].

Most instruction favors the field-independen learner [Reiff, 1996; Davis, 1991]. Field-independent students typically outperform their field-dependent counterparts in all academic subjects [Tinajero and Paramo, 1997]. In none of the reported researches have field-dependent learners outperformed field-independent learners on outcome measures [Davis, 1991].

4.3.2.2 Riding's model of cognitive styles

An example of a "unitary" position is Riding and Cheema's [Riding and Cheema, 1991] point of view. They reviewed over 30 learning style models and concluded that most of the identified styles can be grouped within two independent dimensions (also called "basic cognitive styles" or "meta-styles"; see figure 4.2 based on [Riding and Rayner, 1998]):

- *Wholist-Analytic* — this dimension describes how an individual tends to cognitively *organize* information — either into (w)holes or parts. Wholists tend to form an overall perspective of a situation before delving down into the details, while analytics tend to see the situation as a collection of parts and focus on some of these at a time. (Most psychologists use the term *holist* instead of *wholist*.)

- *Verbalizer-Imager* — this dimension describes how an individual *represents* information while thinking, either as words or mental pictures. For example, verbalizers tend to present information in words, while imagers tend to present information in pictorial form.

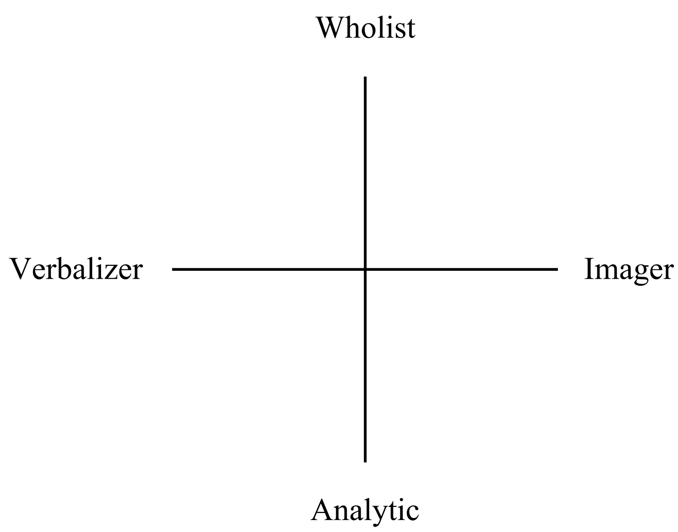


Figure 4.2: Riding's dimensions of cognitive styles

Witkin's FD and FI styles and impulsive/reflective LS of the Dunn and Dunn model can be aligned along the wholist-analytic dimension, whereas the verbalizer-imager and similar verbalizer-visualizer [Richardson, 1997] styles refer to the verbal-imager dimension.

An individual's position on both dimensions can be calculated through a computerized assessment tool called the Cognitive Styles Analysis (CSA) [Riding, 1991a,b,c,d].

4.3.3 Stable personality type

"The instruments and models grouped in this family have a common focus upon learning style as one part of the observable expression of a relatively stable personality type, a theory primarily influenced by the work of Jung [Jung, 1968]" [Coffield et al., 2004]. In this family we present the Myers-Briggs Type Indicator (MBTI).

4.3.3.1 Myers-Briggs Type Indicator (MBTI)

The Myers-Briggs Type Indicator (MBTI) was developed by Katherine Cook Briggs and her daughter Isabel Briggs Myers in 1962. This is the most well-known instrument to assess psychological (personality) type. The model developed by Myers-Briggs involves four different pairs of opposite preferences for how people focus and interact with the outer world:

1. How does a person relate to the world? (either by Extraversion or Introversion):
   - *Extraverts* — try things out, focus on the world around, like working in teams, develop ideas through discussion.
   - *Introverts* — think things through, focus on the inner world of ideas, would rather work alone, ideas come from thinking alone.
2. How does a person absorb/process information? (either by Sensing or Intuition):
   - *Sensors* — concrete, realistic, practical, detail-oriented, focus on facts and procedures, "see the trees instead of forest". Sensing learners like learning facts and solving problems through well established methods. They are generally careful, patient with details, dislike surprises and prefer new knowledge to have some connection to the real world. They have slower reaction to problems, but typically present a better outcome than intuitives.
   - *Intuitives* — abstract, imaginative, concept-oriented, focus on meanings and possibilities, "see the forest instead of the trees". Intuitive learners prefer discovering new relationships and can be innovative in their approach to problem solving. They tend to work faster, not paying much attention to details and dislike repetition and work which involves a lot of memorization and routine calculations. However they are prone to errors and often get lower results than sensing learners.
3. How does a person make decisions? (either by Thinking or Feeling):
   - *Thinkers* — skeptical, tend to make decisions based on logic and rules.
   - *Feelers* — appreciative, tend to make decisions based on personal and humanistic considerations.

4. How does a person manage his life? (either by Judging or Perceiving):
   - *Judgers* — organized, set and follow agendas, make decisions quickly, dislike surprises and need advanced warnings, seek closure even with incomplete data.
   - *Perceivers* — disorganized, adapt to changing circumstances, gather more information before making a decision, enjoy surprises and spontaneous happenings, resist closure to obtain more data.

The standard version of the MBTI asks 93 forced-choice questions (meaning there are only two options) relating to four bipolar discontinuous scales, as shown in figure 4.3):

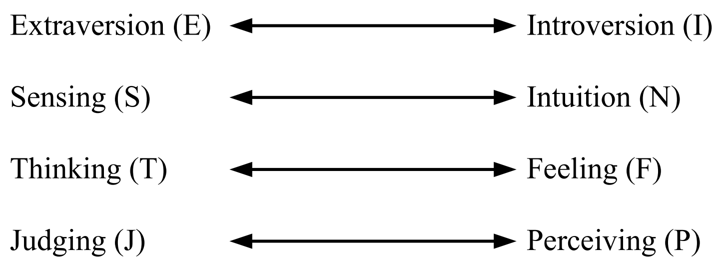


Figure 4.3: The four bipolar discontinuous scales of the Myers-Briggs Type Indicator

The preferences from different pairs can be combined to form 16 personality "types". By taking one preference from each pair a four-letter code is established describing a person, for example, as an ESTP (Extravert, Sensor, Thinker, Perceiver) or an INFJ (Introvert, Intuitor, Feeler, Judger), etc..

[Coffield et al., 2004]: "Some commentators in the learning styles field prefer to exclude the MBTI on the grounds that its scope as a personality measure goes beyond cognitive controls and behavior specifically related to learning. However, the scope of the MBTI includes learning, and it was the authors' intention that it should be a tool to aid learners (Myers, cited by [Tiberio, 1996])".

4.3.4 Flexibly stable learning preferences

In this group, Coffield's report places the models where authors consider that learning style is not a fixed trait, but "a differential preference for learning, which changes slightly from situation to situation. At the same time, there is some long-term stability in learning style" [Kolb, 2000].

We present here Kolb's Learning Style Inventory (LSI), Honey and Mumford's Learning Styles Questionnaire (LSQ) and the Felder-Silverman model.

4.3.4.1 Kolb's Learning Style Inventory (LSI)

Kolb's model is one of the most influential LS models. It was developed by David Kolb in the early 1970s. His theory of experiential learning and the instrument which he devised to test the theory — the Learning Style Inventory (LSI) — have generated a very considerable body of research [Coffield et al., 2004].

According to Kolb [Kolb, 1984]: "learning is the process whereby knowledge is created through the transformation of experience. Knowledge results from the combination of grasping experience and transforming it".

A central principle of his experimental learning theory is a four stage "cycle of learning" that includes the following learning modes:

- *Concrete Experience* (CE — feeling),
- *Abstract Conceptualization* (AC — thinking),
- *Active Experimentation* (AE — doing),
- *Reflective Observation* (RO — watching).

During the learning process the student "touches all the bases". For example, first the learner gets acquainted with the concrete situation, accumulates the experience (CE). This leads to observations and reflections (RO). Further on these are conceptualized into abstract concepts (AC), which the person can actively test and experiment with (AE). As a result this enables the creation of new experiences and the cycle repeats again.

According to Kolb based on the four learning modes there are four basic learning styles, each representing the combination of two learning modes (figure 4.4):

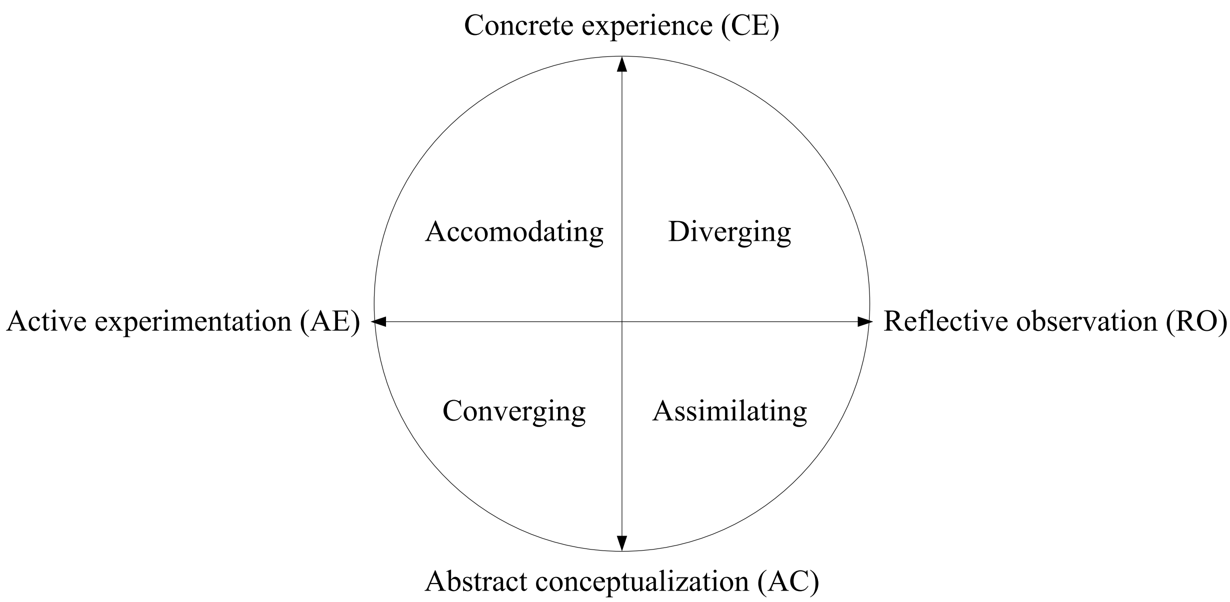


Figure 4.4: Kolb's learning styles model

- *Diverging style* (CE/RO) — the learner with this style emphasizes concrete experience and reflective observation; is imaginative and aware of meanings and values; views concrete situations from many perspectives; adapts by observation rather than by action; interested in people and tends to be feeling-oriented; superior in generating alternative hypothesis and ideas. A characteristic question of this learning type is "Why?". This type of learners responds well to explanations of how course material relates to their experience, their interests, and their future careers.

- *Assimilating style* (AC/RO) — prefers abstract conceptualization and reflective observation. This style is advantageous in inductive reasoning, creating theoretical models. Similar to a person with convergent style this person is more concerned with ideas and abstract concepts than with people. However he tends to focus more on logical soundness and preciseness of the ideas, rather than their practical values. A characteristic question of this learning type is "What?". This type of learners respond to information presented in an organized, logical fashion and benefit if they have time for reflection.

- *Converging style* (AC/AE) — relies primarily on abstract conceptualization and active experimentation; has great advantages at problem solving, decision making and the practical application of ideas; does best in situations like conventional intelligence tests; is controlled in the expression of emotion and prefers dealing with technical problems rather than interpersonal issues. Knowledge is organized through hypothetical-deductive reasoning. A characteristic question of this learning type is "How?". This type of learners responds to having opportunities to work actively on well-defined tasks and to learn by trial-and-error in an environment that allows them to fail safely.

- *Accommodating style* (CE/AE) — emphasizes concrete experience and active experimentation; likes doing things, carrying out plans and getting involved in new experiences; good at adapting to changing circumstances; solves problems in an intuitive, trial-and-error manner; at ease with people but sometimes seen as impatient and "pushy". A characteristic question of this learning type is "What if?". This type of learners likes applying course material in new situations in order to solve real problems.

Among many other correlations between definitions by other theorists, Kolb points out that his "Active Experimentation/Reflective Observation" and "Concrete Experience/Abstract Conceptualization" dimensions correlate with extravert/introvert and feelers/thinkers dimensions of Myers-Briggs Type Indicator (MBTI) respectively.

Kolb's learning styles inventory (LSI) asks individuals to complete 12 sentences that describe learning. (Quoted from [Coffield et al., 2004]): Each sentence (e. g., "I learn best from") has four endings (e. g., AC = "rational theories", CE = "personal relationships", AE = "a chance to try out and practice", and RO = "observation"). Individuals rank the endings for each sentence according to what best describes the way they learn. Four scores, AC, CE, AE and RO, measure an individual's preference for the four modes, and two dimensional scores indicate an individual's relative preference for one pole or the other of the two dialectics, conceptualizing/experiencing (AC-CE) and acting/reflecting (AE-RO).

Kolb does not recommend that the LSI should be used for individual selection purposes because such inventories cannot measure individuals with complete accuracy. Kolb's model has been found to be effective in some language teaching activities.

4.3.4.2 Honey and Mumford's Learning Styles Questionnaire (LSQ)

Peter Honey and Alan Mumford developed their learning styles model in 1982 based on Kolb's theory [Honey and Mumford, 2000]. Kolb's model is more complicated in the sense that its learning styles are defined as the combinations of the learning cycle stages (figure 4.4). Honey and Mumford use different terms for the learning modes of the Kolb model and refer to them as four learning styles (see figure 4.5):

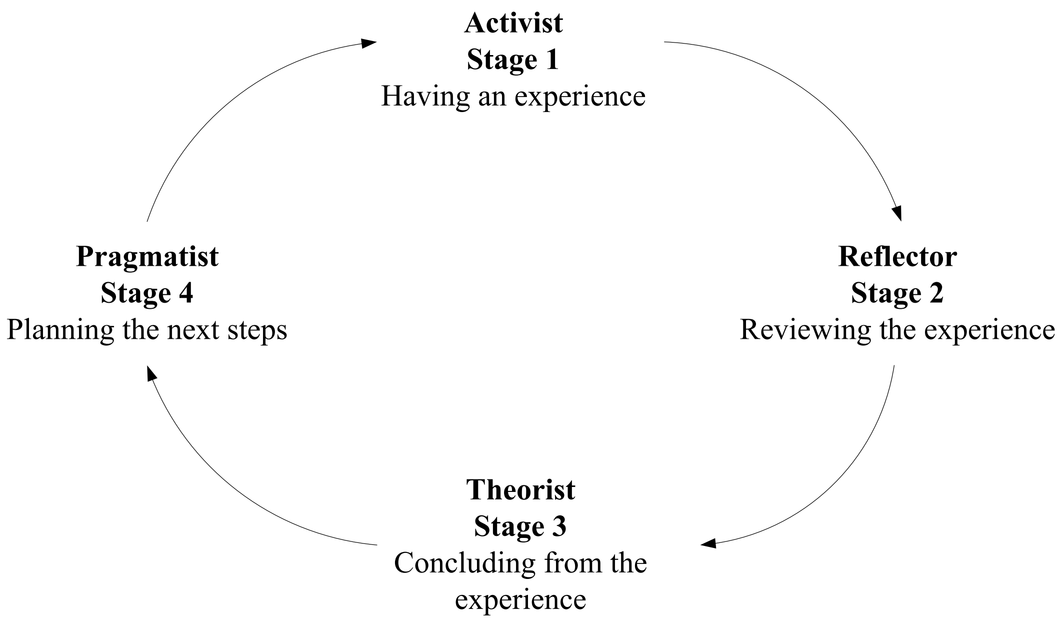


Figure 4.5: Honey and Mumford learning styles model

- *Activist style* (stage 1: "Having an experience", substitution of Kolb's "Concrete Experience") — "I like to have a go and see what happens". Users with this style understand new information by doing something with it, seek challenge and immediate experience, are open-minded. Active learners are keen to try out and experiment with the new information and often enjoy group work because this enables them to do active things.

- *Reflector style* (stage 2: "Reviewing the experience", substitution of Kolb's "Reflective Observation" — "I like to gather information and mull things over". Reflective learners prefer to think about new information first before acting on it, gather data, ponder and analyze, delay reaching conclusions, listen before speaking, thoughtful. They often prefer to first think problems through on their own rather than discussing them in groups.

- *Theorist style* (stage 3: "Concluding from the experience", substitution of Kolb's "Abstract Conceptualization") — "I like to tidy up and reach some conclusions". Theorists think things through in logical steps, assimilate disparate facts into coherent theories, rationally objective, reject subjectivity and flippancy. They learn better through models and concepts, structured programs, and like complex situations to use their skills and knowledge.

- *Pragmatist style* (stage 4: "Planning the next steps", substitution of Kolb's "Active Experimentation") — "I like tried and tested techniques that are relevant to my problems". Pragmatists seek and try out new ideas, are practical, enjoy problem solving and make decisions quickly. They get bored with long discussions.

Honey and Mumford [Honey and Mumford, 2000] mention that "no single style has an overwhelming advantage over any other. Each has strengths and weaknesses but the strengths may be especially important in one situation, but not in another".

They make it clear that they produced their own Learning Styles Questionnaire (LSQ) because they found that Kolb's LSI was not efficient for the managers with whom they worked. Instead of asking people directly how they learn, as Kolb's LSI does — something which most people have never consciously considered — Honey and Mumford give them a questionnaire which probes general behavioral tendencies rather than learning [Coffield et al., 2004].

Both Kolb and Honey and Mumford models are widely known in the LS field and are used extensively in the UK [Coffield et al., 2004].

4.3.4.3 Felder and Silverman's Index of Learning Styles

Felder-Silverman model was first presented in 1988 in [Felder and Silverman, 1988] but was updated since that time.

It involves the following dimensions:

- *Active/Reflective*. [Felder and Silverman, 1988]: "An 'active learner' is someone who feels more comfortable with, or is better at, active experimentation than reflective observation, and conversely for a reflective learner." This dimension is analogous to Kolb's "active experimentation" and "reflective observation" [Felder and Spurlin, 2005] and accordingly to the activist/reflector styles of the Honey and Mumford model. It is also related to extravert and introvert of the Myers-Briggs Type Indicator [Felder and Spurlin, 2005] presented in 4.3.3.1.

- *Sensing/Intuitive*. This dimension is taken directly from MBTI [Felder and Spurlin, 2005] and it is closely related to Kolb's "concrete experience" and "abstract conceptualization" [Felder and Silverman, 1988].

- *Visual/Verbal*. Visual learners prefer visual representations of presented material, such as pictures, diagrams and flow charts. Verbal learners prefer written and spoken explanations.

  Visual/verbal styles of the Felder-Silverman model can be aligned along the imager/verbalizer axis of Riding's model presented in 4.3.2.2.

- *Global/Sequential*. Global learners tend to learn in large jumps, absorbing material almost randomly without seeing connections, and then suddenly "getting it". They may be able to solve complex problems quickly or put things together in novel ways once they have grasped the big picture, but they may have difficulty explaining how they did it. Sequential learners tend to gain understanding in linear steps, with each step following logically from the previous one. They tend to follow logical stepwise paths in finding solutions.

  Global/sequential styles can be aligned along the wholist/analytic dimension of Riding's model.

Richard Felder and Barbara Soloman designed the Index of Learning Styles (ILS)[1] — a 44-question instrument to assess preferences on the four dimensions of the Felder-Silverman model. The Index of Learning Styles is widely used and has been translated into many languages [Zywno, 2003]. It is considered as a suitable psychometric tool for evaluating learning styles of engineering students [Zywno, 2003].

4.3.5 Learning approaches and strategies

[Coffield et al., 2004]: "Researchers within this family refer to underlying personality differences and relatively fixed cognitive characteristics. This leads them to differentiate between styles, strategies and approaches, with the latter being derived from perceptions of a task and cognitive strategies that learners might then adopt to tackle it." Their "view of approaches and strategies — as opposed to styles — takes into account the effects of previous experiences and contextual influences."

4.3.5.1 Pask's model

An influential researcher within this field has been Pask [Pask, 1976] who argues that there are identifiable differences between students' strategies, so that some learners adopt a *holist* strategy and aim from the outset to build up a broad view of the task, and to relate it to other topics and to real-life and personal experience. The opposite strategy is a *serialist* one, where students attempt to build their understanding from the details of activities, facts and experimental results instead of making theoretical connections. Pask makes his holist/serialist distinction from a theory of learning derived from what he calls a conversation between two representations of knowledge.

[1] http://www.engr.ncsu.edu/learningstyles/ilsweb.html

In other words, two distinct strategies Pask identified are:

- *serialists* (partists) — follow a step-by-step learning procedure, concentrating on narrow, simple hypotheses relating to one characteristic at a time;
- *holists* (wholists) — tend to form more complex hypotheses relating to more than one characteristic at a time.

A number of studies have noted that the distinction between field-dependent and field-independent individuals is similar to that of differentiating holists and serialists [Ash, 1986; Brumby, 1982; Entwistle, 1981; Jonassen and Grabowski, 1993; Riding and Cheema, 1991] (cited from [Chen and Macredie, 2002]).

Two tests were invented by Pask to measure serialist and holist strategies: the Spy Ring History Test and the Smuggler's Test. Although Pask's work has been influential in this family of learning styles, both in concepts and methodology, his two tests have not gained credence as reliable or easily usable instruments outside these science disciplines [Coffield et al., 2004].

4.3.5.2 Vermunt's framework for classifying learning styles and his Inventory of Learning Styles (ILS)

For Vermunt, the terms "approach to learning" and "learning style" are synonymous.

Within Vermunt's framework, four learning styles are defined: *meaning-directed*, *application-directed*, *reproduction-directed* and *undirected*. Each is said [Vermunt, 1996] to have distinguishing features in five areas as presented in a 4x5 matrix in table 4.1. It should be noted that the framework is conceived as a flexible one. Vermunt does not claim that his learning styles are mutually exclusive, nor that for all learners the links between areas are always consistent with his theory [Coffield et al., 2004].

Vermunt's framework can be compared with other learning styles models discussed in this chapter.

- *Meaning-oriented* learners prefer to get theory first and then go to examples. This dimension is very similar to the assimilating style of Kolb's model.
- *Application-directed* learners want to know what the information is useful for, later on they develop the theory. This dimension is similar to the accommodating style of Kolb's model.
- *Reproduction-oriented* learners need to know the goals, they try to reproduce the knowledge of experts. They want to get more questions and trial tests. This type is similar to the field-independent style of Witkin's model.
- *Undirected* learners need to be guided. This type is similar to the field-dependent style of Witkin's model.

	Meaning-directed	Application-directed	Reproduction-directed	Undirected
Cognitive processing	Look for relationships between key concepts/ theories: build an overview	Relate topics to everyday experience: look for concrete examples and uses	Select main topics to retain	Find study difficult; read and re-read
Learning orientation	Self-improvement and enrichment	Vocational or "real world" outcomes	Prove competence by getting good marks	Ambivalent; insecure
Affective processes	Intrinsic interest and pleasure	Interested in practical details	Put in time and effort; afraid of forgetting	Lack confidence; fear of failure
Mental model of learning	Dialogue with experts stimulates thinking and engagement with subject through exchange of view	Learn in order to use knowledge	Look for structure in teaching and texts to help take in knowledge and pass examinations. Do not value critical processing of peer discussion	Want teachers to do more; seek peer support
Regulation of learning	Self-guided by interest and their own questions; diagnose and correct poor understanding	Think of problems and examples to test understanding, especially of abstract concepts	Use objectives to check understanding; self-test; rehearse	Not adaptive

Table 4.1. Vermunt's learning styles with illustration of their components (source [Vermunt, 1996])

Vermunt's Inventory of Learning Styles ILS is a 120-item self-rating instrument. Vermunt's model is focused on higher education and is being widely used in Europe. By limiting his focus to higher education, Vermunt has been able to produce a reliable self-assessment tool, but this means that its relevance is largely unknown in other contexts, such as problem-based learning, vocational education, etc..

4.3.6 Discussion

In this section we presented an overview of the learning styles models that involve learning styles having possible implications on adaptive Web-based settings. We have seen that there is much overlap between these models:

- The visual modality of the Dunn and Dunn model is split in two indicating preferences for pictures and text and is therefore correlated with the verbalizer-imager dimension of Riding's model and the verbal-visual dimension of the Felder-Silverman model.

- The global/analytic styles of Dunn and Dunn's model, Riding's wholist/analytic dimension, the global/sequential styles of the Felder-Silverman model, Pask's (w)holists-serialists — "[Clarke, 1993] suggests that all these styles 'differ more in name than nature'" (quoted from [Bajraktarevic et al., 2003]).

  Indeed, if we look at the descriptions of these LS we can actually see that they indicate similar tendencies of approaching information. Global/(w)holist styles reflect a tendency to adopt a relatively top down approach. On the other hand, analytic/sequential/serialist styles reflect a tendency to adopt a relatively bottom up approach.

- The active/reflective dimension of the Felder-Silverman model is similar to the activist/reflector styles of the Honey and Mumford model and the extraverts/introverts of MBTI (see 4.3.4.3).

- The sensing/intuitive dimension of the Felder-Silverman model is taken directly from MBTI and is related to its sensors/intuitives (see 4.3.4.3).

- The styles of Vermunt's model are correlated with Kolb's and Witkin's models as follows (see 4.3.5.2):

  - meaning-oriented is similar to the assimilating style of Kolb's model,
  - application-oriented is similar to the accommodating style of Kolb's model,
  - reproduction-oriented is similar to the field-independent style of Witkin's model,
  - undirected is similar to the field-dependent style of Witkin model.

We visualize the correlations between the learning styles of the discussed models in figure 4.6 and will be referring to it later.

4.4 Learning Styles Assessment

All learning style models presume that it is possible to measure the learning styles they involve. Each provides its own psychometric instrument to assess users' learning styles. Some of the LS questionnaires put a user directly into a certain stereotype (e. g., Kolb's model), some use strong, low, etc. preferences (e. g., the Dunn and Dunn model and the Felder-Silverman model).

Coffield's report [Coffield et al., 2004] raises the following questions concerning the status of the learning style inventories:

- *Reliability* — Do they measure the learning styles of the students consistently?

- *Validity* — Is it really a test of learning styles or of some other quality such as intelligence or personality?

There are several problems with respect to existing questionnaires. Coffield's report mentions "In many ways, the use of different inventories of learning styles has acquired an unexamined life of its own, where the notion of learning styles itself and the various means to measure it are accepted without question." "Some learning style theorists have conducted repeated small studies that tend to validate the hypotheses derived from their own conceptualizations. However, in general, these studies have not been designed to disconfirm hypotheses, are open to expectation and participation effects, and do not involve

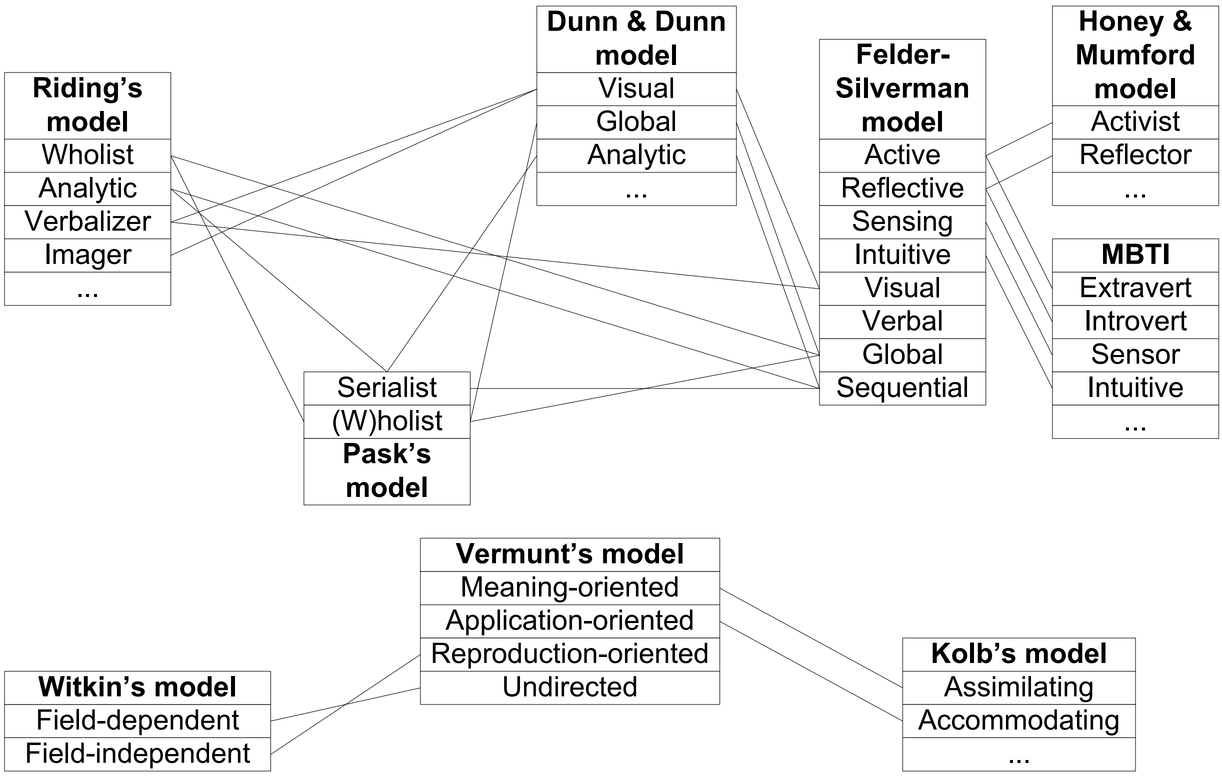


Figure 4.6: Correlations between learning styles

wide enough samples to constitute valid tests in educational settings. Even with these built-in biases, no single learner preference pattern unambiguously indicates a specific instructional design."

Another aspect of complexity is that "researchers produce their models and instruments for different purposes. Some aim to contribute to theory about learning styles and do not design their instrument for use in mainstream practice. In contrast, others develop an instrument to be used widely by practitioners in diverse contexts. This difference affects the type of claims made for the instrument and the type of research studies that evaluate it."

Moreover, the problem with assessing learning styles is that some of them may vary from context to context or even from task to task [Coffield et al., 2004], or depending on the intellectual load, instruction, etc. [Holodnaya, 2002].

Coffield's report showed that some of the best known and widely used instruments have serious weaknesses (e. g., low reliability, poor validity). Each of the 13 models presented in the report was examined for evidence, provided by independent researchers, test-retest reliability, construct and predictive validity that the instrument could demonstrate both internal consistency[2] and test-retest reliability[3] and construct validity[4] and predictive validity[5]. The results for some of the models discussed in this dissertation were as follows:

- Dunn and Dunn model met predictive validity criteria;
- Riding's model did not meet a single criterion;
- Myers-Briggs met internal consistency and test-retest reliability;
- Kolb and Honey and Mumford models met test-retest reliability criteria;
- Vermunt model met internal consistency, test-retest and construct validity criteria.

[2] "Definition taken from http://en.wikipedia.org/wiki/Internal_consistency: In statistics and research, internal consistency is the coefficient of test scores. Put more simply, internal consistency measures the consistency of results across items within a single test. One example of internal consistency in research might be a test of two questions. The first statement says "You feel negatively about bicycles." The second statement would say "You like to ride bicycles." If a person agrees with the first and disagrees with the second, the test has internal consistency."

[3] Definition taken from http://www.statistics.com/resources/glossary/t/trtreliab.php: "The test-retest reliability of a survey instrument, like a psychological test, is estimated by performing the same survey with the same respondents at different moments of time. The closer the results, the greater the test-retest reliability of the survey instrument. The correlation coefficient between such two sets of responses is often used as a quantitative measure of the test-retest reliability."

[4] [Felder and Spurlin, 2005]: "Construct validity signifies the extent to which an instrument actually measures the theoretical construct or trait that it purports to measure."

[5] Definition taken from http://en.wikipedia.org/wiki/Predictive_validity: "In psychometrics, predictive validity is the extent to which a scale predicts scores on some criterion measure. For example, the validity of a cognitive test for job performance is the correlation between test scores and, say, supervisor performance ratings. Such a cognitive test would have predictive validity if the observed correlation were statistically significant."

The report shows that despite being refined and revised for over 30 years, many models poorly satisfy the criteria. One of the conclusions of the report was the marked variability in quality among the 13 examined LS models; they are not all alike nor of equal worth and it matters fundamentally which instrument is chosen.

Filling in psychological questionnaires is a time consuming process because all of them are quite long. Moreover, as was discussed above, the results are not always valid and reliable and LS may vary under various conditions. This brings us to the idea that if we would like to consider LS in adaptive hypermedia but do not want the users to do much extra work, we would need to avoid (long) questionnaires. Instead we could enrich the system with the possibility of unobtrusively detecting certain patterns in the user's browsing behavior that might correspond to a particular LS. In that case the system could also detect that the user's browsing patterns are different in various contexts and could provide adaptation accordingly.

4.5 Conflicting Ideas About Learning

When considering effective learning, one cannot ignore the importance of such factors as a learner's intelligence, prior knowledge of the subject domain, level of motivation, anxiety, self-confidence, and the amount of learner control offered. The research in the field of learning styles suggests that another important factor with regard to this issue is the student's cognitive or learning style [Riding and Rayner, 1998].

One of the most popular recommendations from psychologists is that the learning styles of the students should be linked to the teaching style of the tutor, the so-called "matching hypothesis". From the first glance it seems logical that this would increase the learner's performance.

Felder [Felder, 1993] mentions that mismatching can have serious consequences. Students can feel "as though they are being addressed in an unfamiliar foreign language. They tend to get lower grades than students whose learning styles are better matched to the instructor's teaching style and are less likely to develop an interest in the course material". This is especially worrying if the mismatches are extreme. Felder complains about the negative outcomes of unintentional mismatching where, for instance, teachers are unaware of their own learning style and may, as a result teach only in that style, thus favoring certain students and disadvantaging others [Coffield et al., 2004].

Dunn and Dunn [Dunn, 2003b] suggest that teachers adapt the instruction and environmental conditions by allowing learners to work with their strong preferences and to avoid, as far as possible, activities for which learners report having very low preferences. People who have no high or low preferences do not need "matching" and can therefore adapt more easily to different teaching styles and activities.

One problem with this approach is that if the teacher provides only a certain type of material for a student with a particular style, for example, textual descriptions for verbalizers and images and diagrams for visualizers, it will not be possible to create an adequate picture of the situation [Holodnaya, 2002]. Moreover some researches claim that constant repetition of the learning cycle — for example, beginning every new task with concrete experience — could quickly become tiresome. Grasha [Grasha, 1984] asks: "How long can people tolerate environments that match their preferred learning style before they become bored?" Gregorc's research [Gregorc, 1984] supports Grasha's point of view and suggests that even those individuals with strong preferences for a particular learning style preferred a variety of teaching approaches to avoid boredom. Grasha's aim would be to "teach people new learning styles or at least let them sample unfamiliar ones". Holodnaya [Holodnaya, 2002] also mentions that for more able users, mismatching materials to cognitive styles may be advantageous as it encourages users to develop new learning strategies that would cope with a wider range of materials and experiences in the future.

A number of experiments have been carried out to find out the relationship between matching/mismatching learning material to LS and the students performance.

Pask and Scott's research [Pask and Scott, 1972] on matching of teaching material to the holist/serialist style showed that students learn faster and more effectively where a match occurs. Ford [Ford, 1985, 1995; Ford and Chen, 2001] based his experiments with FD/FI learners on Pask's work on holist/serialist dimension (taking into account the similarity of behavior of the students with these styles). He provided a breadth-first/depth-first structure of the learning material to FD/FI students as Pask did for holists/serialists. Three conducted studies concluded that matching was linked with improved performance. Still his most recent study [Ford and Chen, 2001] suggests that the effects of matching and mismatching "may not be simple, and may entail complex interactions with other factors such as gender, and different forms of learning".

However, critics highlight that only a limited number of studies have demonstrated that students learn more effectively when their learning style is accommodated [Holodnaya, 2002]. A large number of experiments with various learning styles did not show any significant difference [Holodnaya, 2002; Hayes and Allison, 1993]. Coffield's report [Coffield et al., 2004] mentions that there is no evidence that matching of teacher style with the learning styles measured by the Kolb and Myers-Briggs Indicator (MBTI) has any positive effect on achievement.

(Quoted from [Coffield et al., 2004]): Smith, Sekar and Townsend [Smith et al., 2002] recently researched the impact of surface and reflective teaching and learning on students' academic success and found nine studies which showed that learning is more effective where there is a match and nine showing it to be more effective where there is a mismatch. They concluded : "For each research study supporting the principle of matching instructional style and learning style, there is a study rejecting the matching hypothesis".

Mitchell, Chen and Macredie [Mitchell et al., 2004] describe the results of their experiment investigating the learning performance and user-perceptions of FD/FI learners using educational adaptive hypermedia interfaces. Participants used two interfaces — one "normal" and one adaptive — and were randomly matched or mismatched to their cognitive styles. The studies concluded that whilst a wrongly adapted (mismatched) interface may cause problems for some users, appropriately adapted (matched) interfaces may be no more effective than a well-designed interface for all users. The results suggest that adapting to an interface that is too specific may restrict the users, who may not necessarily prefer all aspects of the interface that are considered to be useful for a user with that cognitive style. The authors propose that it is possibly more beneficial for hypermedia system designers to concern themselves with an interface that is easy (and not too restrictive) to use for all users, regardless of their level of field-dependence. Trying to create distinct interfaces for different levels of field dependence may do more harm than good.

However, as the authors themselves mention, this study was limited in that it provided adaptation to FD/FI in a way considered appropriate for such individuals based on interpretations of previous research into field-dependence and hypermedia learning. For future work they consider re-interpretation of an ideal interface for FD/FI learners.

Moreover this experiment was restricted to the study of field-dependence as measured by Cognitive Styles Analysis (CSA by Riding). Future research should also strive to determine whether the findings from this study regarding adaptation to field-dependence apply to other cognitive styles and other cognitive style assessment tools.

Brown, Stewart and Brailsford [Brown et al., 2006] carried out experiments to see if matching/mismatching to visual and verbal styles of the learners would make any difference. Although the user trials did not indicate any statistically significant differences, qualitative information gleaned from the study indicated that students preferred using this environment over other traditional revision methods, that they perceived personalized tuition to be better than a "one size fits all" approach.

From the results presented above we see that there is no definite answer to the question whether the instruction should be matched to the student's style or not. Coffield's report [Coffield et al., 2004] draws a conclusion that "it is simply premature to be drawing simple implications for practice when there is so much complexity and so many gaps in knowledge". Another argument against any large-scale adoption of matching is that it would be very hard for teachers to accommodate their teaching methods to a variety of learning styles of their students. And even if the teacher has chosen only a few learning styles then repeating the instruction with the same style would become tiresome.

While there are doubts as to the validity of using LS, we still consider that this is a useful endeavor. We consider that, first, it is good for the learner to be aware of his/her learning style and to know what his/her strengths and weaknesses are. Second, it is important to provide the learner with the variety of instructional strategies and let him/her choose the one (s)he prefers. Furthermore, as has been mentioned above, intentional mismatch may help more advanced learners develop new skills.

Another important topic to be researched is — "teaching styles". It is appropriate to assume that the cognitive style of the designer (or teacher) may influence the way in which they design their applications. How to foresee this possible problem of "style conflict" between a teacher's and a learner's styles?

The topic of "teaching styles" has its own literature, theories and controversies, but it is beyond the scope of this dissertation. It is sufficient to refer here to the myriad of interactions between the learning style of the student and the objectives, content, sequence, teaching methods and social context of the lesson. We consider that designers should know about the peculiarities of their own styles.

4.6 Summary

In this chapter we provided a review of the learning styles theory. We discussed the most popular and influential LS models:

- Dunn and Dunn model;
- Witkin's model;
- Riding's model;
- Myers-Briggs Type Indicator;
- Kolb's Learning Style Inventory;
- Honey and Mumford's Learning Style Questionnaire;
- Felder and Silverman's Index of Learning Styles;
- Pask's model;
- Vermunt's model.

We can see from the descriptions of these models that there is much overlap between them. We consider these similarities to discuss possible implications in educational Web-based settings in the following chapter.

Currently there is no general consensus whether it is reasonable or not to apply cognitive/learning styles in pedagogy in order to improve the learner's outcome (see section 4.5). However, we consider that first, it is good for the learner to be aware of his/her learning style and to know what his/her strengths and weaknesses are. Second, it is important to provide the learner with a variety of instructional strategies and let him/her choose the one (s)he prefers. Third, intentional mismatch may help more advanced learners develop new skills.

Coffield's report raises some concerns about the reliability and validity of some of the LS instruments (see section 4.4). We consider that it would be good to foresee the mechanism for inferring users' preferences that might correspond to certain LS.

The theory and practice of learning styles provides us with new interesting ideas for our further research. It concerns providing a new type of adaptive behavior of our system that can be based on the recommendations from psychologists concerning adapting instruction to a particular learning style. We also consider that the developed tools can contribute to the psychological research by providing a test bed for the learning styles theories.

Chapter 5

Connections Between LS and AH

This chapter investigates the possible ways of representing learning styles in adaptive hypermedia environments. The questions being addressed in this chapter are: Can the learning material be presented in a way that suits the preferred students' approach to learning? What should be the format of learning materials to fit a certain learning style? Is there a single and "correct" way to represent a particular learning style in adaptive hypermedia? Can learning styles be authored in current Web-based systems? What are the main issues when authoring learning styles? Is it possible to detect learning styles in a Web-based environment?

Section 5.1 provides recommendations from psychological research for possible teaching methods (also called "instructional strategies" or "instructional design" through the dissertation) to address certain LS. In section 5.2 we investigate how other researchers have incorporated LS into their adaptive hypermedia systems. It gives examples of existing AHS providing adaptation to learning styles. It also compares our approach (discussed in detail in chapter 6) with the ones adopted in the presented systems. In section 5.3 we identify the types of strategies for representing learning styles in adaptive Web-based environments. Section 5.4 summarizes the chapter and provides an answer to research question 2 of this dissertation "Can we specify how an AHS should perform adaptation according to different learning styles?".

5.1 Implications of LS for Pedagogy

Learning styles have been intensively studied in the classical educational (classroom) settings. In such an environment a teacher is able to identify learners' individual differences with regard to learning styles and thus provide them with individually selected and structured learning material. However, it is much more difficult to provide such interaction with individual learners in a Web-based environment. One of the problems consists of identifying the learner's preferences or learning styles. How can the system the user interacts with obtain this information? Another question is: once the learner's preferences or learning styles are known what should the system's adaptive response be? In this section we address the latter issue.

Though application of LS to an adaptive course design still seems to be problematic, for (some) LS outlined in chapter 4 we found several resources with recommendations from psychologists for a possible format of the learning material and the instructional design. We present an overview of LS with their corresponding implications for pedagogy in table 5.1 and provide a more detailed description of instructional strategies here.

Learning styles reflecting sensory preferences

The Dunn and Dunn model describes the following sensory modalities: auditory, visual, tactile and kinesthetic. The visual modality is split in two indicating preferences for pictures and text and as figure 4.6 shows is correlated with the verbalizer-imager dimension of Riding's model and the verbal-visual dimension of the Felder-Silverman model.

Riding and Rayner [Riding and Rayner, 1995] suggest that the teacher should provide written material to the verbalizers and pictorial presentations to the visualizers (such as pictures, diagrams, charts and graphs). (Verbalizer versus visualizer is a term similar to verbalizer versus imager and was proposed by Richardson [Richardson, 1997].) According to [Riding and Cheema, 1991; Riding and Douglas, 1993] the imagers perform better in a "text-plus-picture" condition, whereas the verbalizers are better in a "text-plus-text" condition (text is illustrated or elaborated with more text). In table 5.1 we show the same recommendations for all correlated styles.

Similar to the previous example, auditory students can be provided with more audio material. Interactive multimedia elements ("interactivelets"): puzzles, drag & drop fill-ins, small games should be provided to the tactile and kinesthetic learners [Rundle and Dunn, 2000].

In terms of adaptive hypermedia technologies these types of adaptation can be realized through adaptive presentation support (or content adaptation).

Global/(w)holist and analytic/sequential/serialist styles

Figure 4.6 shows that Pask's wholists are correlated with the global style of the Dunn and Dunn, Riding and Felder-Silverman models, whereas serialists are correlated with the analytic style of the Dunn and Dunn and Riding's models and the sequential style of Felder-Silverman model. Series of Pask's experiments with wholists and serialists in 1970s showed that material structured in breadth-first order maps well to the strategies adopted by wholists whereas depth-first navigational paths suit serialists. These results can be taken into consideration while designing an application for learners with the LS grouped in this sub-section.

A breadth-first and depth-first structure of an online course can be realized through adaptive navigation support. For breadth-first structure, after a learner has read (something about) a recommended concept (topic) the next recommended places to go to will be the concepts at the same level in the course hierarchy, and only after the learner has

Learning styles	Recommendation for pedagogy
Auditory	More audio material
Visual/Visualizer/ Imager	More pictorial material, such as diagrams, charts, graphs (possibly plus text)
Verbal/Verbalizer	More textual material (possibly plus audio)
Tactile/Kinesthetic	Interactive elements: puzzles, drag & drop fill-ins, small games
Global/(W)holist	Breadth-first structure of the material (possibly guided navigation at each level)
Analytic/Sequential/Serialist	Depth-first structure of the material
Field-dependent/Undirected	Breadth-first structure of the material, guided navigation, route suggestion with instructional cues or background colors, context highlighting Illustration of the ideas with visual materials Graphic visualization: main menu, concept map or graphic path indicator Advance organizer Program control Providing maximum feedback Providing social features
Field-independent/ Reproduction-oriented	Depth-first structure of the material or navigational freedom Index, search option Learner control Provide individual environment
Activist/Active/ Extravert	Activity-oriented approach: showing content of activity and links to example, theory, exercise
Reflector/Reflective/ Introvert	Example-oriented approach: showing content of example and links to theory, exercise, activity
Pragmatist	Exercise-oriented approach: showing content of exercise and links to example, theory, activity
Theorist	Theory-oriented approach: showing content of theory and links to example, exercise, activity
Converger	Abstract information
Diverger	Concrete information
Assimilator/Meaning-oriented	Theory-based information
Accomodator/Application-oriented	Application-based information
Sensing/Sensor	Examples before expositions
Intuitive	Expositions before examples
Judger	Rich media, such as pictures, tables, and diagrams
Perceiver	Lean media (without the integrative use of pictures, tables and diagrams)

Table 5.1: Recommendations for implications of LS in pedagogy

read all of them the concepts at a lower level are recommended. For depth-first structure, after a learner has read a recommended concept the next recommended concepts will be concepts at a lower level. This process is repeated recursively until the leaf concepts are reached. A sibling of a concept would be recommended only after this concept and all the concepts in its subtree have been read.

For a breadth-first structure, we would suggest having a choice of either providing completely guided navigation at each level (meaning that when a recommended concept has been read only its next sibling is recommended) or providing navigational freedom at each level (meaning that all concepts at a certain level are recommended after their parent concept has been read). Furthermore we would suggest having a possibility of providing breadth-first and depth-first structures only up to a certain level, meaning that after reading the concepts at this level all lower level concepts are recommended to every type of learner. Whether the learner wants a global view of a course before studying details or not, at some point a topic is selected (which is located at some depth in the concept tree) and from there on the subtree of small concepts covering that topic must be studied by every learner. The difference between global or (w)holist and others no longer matters at the level of detailed topics.

Field-dependent/field-independent styles of Witkin's model and undirected and reproduction-oriented learners of Vermunt's model

Among the various dimensions of cognitive styles FD/FI is probably the most widely studied one, with the broadest application to problems in education. Researchers provide a number of suggestions concerning the navigation support and user interface (layout) for FD/FI learners.

Taking into consideration the global approach to learning of FD learners versus analytical approach of FI ones, Ford, Chen, Macredie [Ford, 1995; Ford and Chen, 2001; Chen and Macredie, 2002] based their research on the FD/FI dimension on Pask's experiments with holists and serialists. Their findings suggest that, similarly to holists, FD learners should be provided with a breadth-first structure in hypermedia learning systems and, similarly to serialists, FI ones should be provided with depth-first structure. FD learners may have more difficulties in "non-linear" learning; they may get confused more easily than those with a strong FI tendency. FD learners are more likely to require externally defined goals, maximum amount of guidance is required for them. On the other hand FI learners tend to develop self-defined goals and reinforcements, set learning paths themselves; minimal guidance and direction are needed for them. Taking this into account, we would suggest providing FI learners with a freedom of navigation instead of forcing them to navigate in depth-first order. However we consider that this choice should be left to the authors designing their adaptive courses for FD/FI learners.

Furthermore to minimize disorientation problems for FD learners Chen and Macredie [Chen and Macredie, 2002] suggest usage of the following techniques that affect the user interface:

- highlighting context — proper use of font sizes and colors may facilitate FD learners to identify the part of the information being explored and the relative position in context. In addition, screen elements of the user interface should form a well-organized layout that draws attention to the important pieces.

- effective feedback — e. g., includes use of different navigation buttons, showing progress on each page (e. g., "Page 4 of 10").

- graphic visualizations — maps or hierarchical diagrams showing current location of the learner. In addition, some identical symbols (e. g., a check mark) can be shown on the map to serve as feedback to indicate pages visited. Furthermore, pop-up windows can be applied to present history-based annotations that outline links to previously visited pages [Eklund et al., 1997].

Graphic visualizations, as well as a main menu that show the whole picture of the context can serve as navigational tools to support global learning. They help FD learners to organize information and act as a checklist to foster learning. Also a study by Meng and Patty [Meng and Patty, 1991] showed that FD learners benefit most from advanced organizers that serve as a bridging strategy, which provides a connection between one unit and another [Triantafillou et al., 2002]. On the other hand, FI students who prefer to find specific information in hypermedia learning systems need to have navigation tools that can directly facilitate finding and jumping to the location of particular items. Index, query searching, or other tools that allow more analytical approaches and active engagement can support FI students in an efficient way of learning.

In terms of control features, there is evidence that FI learners prefer to be left in control of the system, while FD learners are more comfortable relinquishing control to the system [Hedberg and McNamara, 1989; Yoon, 1993]. These findings can be considered for providing FI learners with a learner control option while providing a program control option for FD ones.

Furthermore Chen and Macredie propose providing the following options:

- successive options — providing a series of options, learners can select from based on their own preferences. For example, a simple factual screen can be followed by displaying a pop-up window to show more expansive information if the FD learners need it.

- switch of cues — in a large volume of textual information, the learning program should provide a framework of navigational cues for FD learners. However, such cues can be switched off, lest they irritate FI learners.

Moreover, [Triantafillou et al., 2002] suggests providing social features versus an individual environment respectively for FD and FI learners. Thus FD learners should be strongly recommended to use such collaborative tools as chat or discussion forums, whereas these options may be not necessary for FI ones.

We would suggest providing the same type of adaptive behavior as for field-dependent and field-independent learners for undirected and reproduction-oriented learners of Vermunt's model respectively, given the similarity between the dimensions (see figure 4.6).

Honey and Mumford model styles, active/reflective dimension of Felder-Silverman model and extraverts/introverts of the Myers Briggs Type Indicator

Papanikolaou et al. [Papanikolaou et al., 2003] propose the following instructional strategies for the learning styles described in the Honey and Mumford model:

- activity-oriented with high interactivity level for activists, who are more motivated by experimentation and challenging tasks;
- example-oriented for reflectors who tend to collect and analyze data before taking action;
- exercise-oriented for pragmatists, as they are keen on trying out ideas, theories and techniques;
- theory-oriented for theorists, giving them the chance to explore and discover concepts in more abstract ways.

According to the proposed approach, all learners are provided with the same *knowledge modules* — multiple representations of the concepts being studied, such as theory presentations (definitions, descriptions, conclusions), questions introducing or assessing the concept, examples (concrete instantiations of concepts, application examples, analogies), exercises, activities (activities using computer simulation, exploration activities, case studies), definitions in the glossary, etc.. However, the method and order of their presentation is adapted, according to different instructional strategies that focus on different perspectives of the concepts. The various knowledge modules are presented in different areas of an educational material page, and they are either embedded in the page, or appear as links.

In particular, Papanikolaou et al. [Papanikolaou et al., 2003] propose that an activist starts with an activity and the system then provides him/her with all necessary information. The knowledge module "activity" will be embedded at the beginning of the page while the rest of the modules (example, theory, exercise) will appear next as links.

A reflector is recommended to start reading example, continue with a brief theory presentation and then try to solve an exercise. Concerning the implementation of the adopted instructional strategy, the knowledge module "example" will be embedded in the beginning of the page while the rest of the modules (theory, exercise, activity) will appear next as links.

We propose the same type of adaptation as for the activist/reflector styles of the Honey and Mumford model, for the active/reflective dimension of the Felder-Silverman model and the extraverts/introverts of MBTI, given the similarity between these dimensions (see figure 4.6).

Similarly, the pragmatist starts with an exercise and continues with links to examples, theory hints and activities. The theorist starts with theory hints and continues with links to examples, exercises and activities.

Such kind of adaptive behavior can be provided using adaptive presentation techniques. Sequencing of the knowledge modules can also be done using adaptive navigation support.

Kolb's model styles and meaning-oriented and application-oriented learners of Vermunt's model

According to Liu and Ginther [Liu and Ginther, 1999] the teacher should selectively provide abstract information to the convergers and concrete information to the divergers, theory-based learning to the assimilators and application-based learning to the accomodators. Similar to the previous example for LS described in the Honey and Mumford model, the teacher can have a number of knowledge modules describing concepts and present these concept for learners with different styles from different perspectives through adaptive presentation or adaptive navigation support.

We propose the same type of adaptation as for the assimilating and accomodating styles of Kolb's model for meaning-oriented and application-oriented learners of Vermunt's model respectively, given the similarity between these dimensions 4.3.5.2.

Sensing-Intuitive dimension of the Felder-Silverman model and the Sensor/Intuitive dimension of the Myers Briggs Type Indicator

Paredes and Rodrigues [Paredes and Rodrigues, 2002] propose an adaptation procedure for moderate and strong sensing-intuitive learners as classified in the Felder-Silverman model. In their approach a course is described in terms of Teaching Tasks (TT) and rules. Knowledge is represented by means of TTs that need to be achieved. TTs may be exposition tasks, practical or examples. Adaptation lies in presenting examples before expositions to moderate and strong sensing learners and the opposite to moderate and strong intuitive learners. Similar to the previous examples of Honey and Mumford's and Kolb's model, this functionality can be realized through adaptive presentation or adaptive navigation support.

We propose the same type of adaptation for the sensor/intuitive dimension of the MBTI model given its similarity with the sensing-intuitive dimension of Felder-Silverman model (see 4.3.4.3).

The Perceptive-Judging dimension of the Myers Briggs Type Indicator (MBTI)

According to Liu and Ginther [Liu and Ginther, 1999] perceptive learners are expected to prefer rich media, such as the integrative use of pictures, tables and diagrams, while judging learners are expected to prefer lean media (without the integrative use of pictures, tables and diagrams). Adaptive response to these preferences can be realized through adaptive presentation techniques.

5.2 Examples of Existing AHS Providing Adaptation to LS

Several systems which provide adaptation to users' learning styles exist. Table 5.2 presents some of these systems, the learning styles they implement, the methods for identifying these LS and the provided types of adaptation. We discuss these issues in more detail in the following sub-sections.

5.2.1 Provided types of adaptation

In this section we group the systems with support for learning styles based on the similarity of the provided types of adaptation.

ACE, INSPIRE, Tangow

In ACE (Adaptive Courseware Environment), INSPIRE (INtelligent System for Personalized Instruction in a Remote Environment) and Tangow (Task-based Adaptive learNer Guidance On the Web) adaptation lies in presenting a different sequence of alternative contents for the concepts. ACE is not bound to a certain learning style model. It can respond to various learner's preferences for sequencing of learning materials that might correspond to different learning styles. INSPIRE provides adaptation to learning styles described by the Honey and Mumford model, Tangow — to the sensing-intuitive dimension of the Felder-Silverman model. Concepts are represented by "example", "activity", "theory", "exercise" in INSPIRE and by "example", "exposition" in Tangow. For example, for reflectors in INSPIRE and sensing users in Tangow the instructional strategy is example-oriented, meaning that the learners are presented with an example first and only afterwards with other representations of the concept. In INSPIRE, to be precise, this means that the system will show the content of an example followed by links to theory, exercise and activity.

iWeaver, CS388, MANIC

In iWeaver, CS388 and MANIC the adaptation is achieved by providing different media representations for each learner (type).

iWeaver adopts the learning styles described in the Dunn and Dunn model. The following media types are presented for different types of learners: textual information for learners with a preference for text; power-point style presentations and streaming audio for auditory learners; multimedia presentations created with Flash — animations, puzzles, drag and drop fill-in examples, little games and riddles — for kinesthetic and visual learners; a "try-it" button for impulsive learners; context-aware note taking tool for reflective learners.

System	Learning styles	Method of learning style identification	Provided types of adaptation
ACE [Specht and Opperman, 1998]	Preferences for sequencing of learning materials	Introductory questionnaire, monitoring which sequences of learning material are requested	Sequencing of learning materials according to a particular teaching strategy (learning by example, reading texts or learning by doing), based on learners interests and material preferences
AES-CS [Triantafillou et al., 2002]	FD versus FI	Group Embedded Figures Test (GEFT) questionnaire [Witkin et al., 1971]; learners can later on change their learning style through the Learner Model	Program vs. learner control, use of contextual organizers (advance vs. post), instructional support, navigational tools, feedback to assessment questions
APeLS [Canavan, 2004]	VARK learning styles, Kolb's, Honey and Mumford LS model (in principle possible to implement any LS model)	Assessing LS with corresponding psychometric questionnaire	Adaptive presentation and adaptive navigation support through selection and sequencing of candidates
CAMELEON [Laroussi and Benahmed, 1998]	Learning styles of the Felder-Silverman LS model	Felder-Soloman ILS (Index of Learning Styles) questionnaire	Adaptive presentation through page variants and adding prerequisites, adaptive navigation support through global guidance with link removal
CS388 [Carver et al., 1996]	Global/sequential, visual/verbal, sensing/intuitive dimensions of the Felder-Silverman LS model	Felder-Soloman ILS (Index of Learning Styles) questionnaire	Lesson media elements (such as slideshows, digital movies, sound, text) are presented in a sorted list ranked from the most to least conducive based on learners' LS

INSPIRE [Grigoriadou et al., 2001]	Activist, pragmatist, reflector and theorist learning styles of Honey and Mumford LS model	Honey and Mumford questionnaire	Adaptive presentation through different sequencing of concept representations such as “activity”, “example”, “theory”, “exercise”; showing the content of one concept representation and only links to other representations
iWeaver [Wolf, 2002]	Learning styles of Dunn and Dunn LS model	Building Excellence Survey [Rundle and Dunn, 2000] as an assessment tool; the system allows learners to try different media representations intended for different learning styles	Textual information for learners with preference for text; Power-point style presentations, streaming audio for auditory learners; Multimedia presentations created with Flash: animations, puzzles, drag and drop fill-in examples, little games and riddles for kinesthetic and visual learners; “Try-it” button for impulsive learners; Context-aware note taking tool for reflective learners
MANIC [Stern and Woolf, 2000]	Applies preferences for graphic versus textual information	Determining learner’s preferences in terms of explanations, examples and graphics	Adaptive presentation through stretchtext technique; allows graphics, as well as text, to be used as supplemental information
Tangow [Carro et al., 1999]	Sensing-intuitive dimension of learning styles from the Felder-Silverman LS model	Felder-Soloman ILS (Index of Learning Styles) questionnaire	Adaptive presentation through different sequencing of concept representations such as “example”, “exposition”

Table 5.2: Learning styles incorporated into adaptive systems

In CS388 a learner is presented with a list of media elements (such as slideshows, graphics, digital movies, text) sorted from most to least conducive based on their effectiveness to his/her learning style.

Based on a learner's preference for graphic versus textual information MANIC provides adaptive presentation through the stretchtext technique allowing graphics, as well as text, to be used as supplemental information. The authors color the keyword being "stretched" and the supplemental information added after the sentence in which the keyword appears, so that the learner knows which supplemental material goes with which keyword. The learner is also given a link allowing to hide the information that is being shown.

CAMELEON

CAMELEON (Computer Adaptive MEdium for LEarning On Networks) provides adaptation to the learning styles of the Felder-Silverman model. Via CAMELEON course material can be split either in composite or elementary fragments (image, video sequence, text, etc.). Each fragment can be tagged with values indicating the category of learners (visual/verbal, active/reflective, sensing/intuitive, sequential/global). The page generator will then deliver the suitable course to the learner. CAMELEON adds prerequisite explanations and provides page variants, thereby achieving content adaptation. Furthermore CAMELEON adapts the hypergraph of the course by removing the links from that graph that are not suitable for the learner. Through global guidance with link removal it achieves adaptive navigation support.

APeLS

Both selection and sequencing mechanisms are applied to provide adaptation to learning styles in APeLS (Adaptive Personalised e-Learning Service). APeLS applies VARK (Visual, Auditory, Read/Write and Kinesthetic) learning styles and Kolb's model for selecting content for the learner. Selection mechanisms (called "candidate selectors" in APeLS terminology) can be implemented in two modes — all (set of candidates that meet the minimum requirements) and best (single candidate that best fits the requirements). When there are several candidate selectors each one called refines the list of candidates with the last one called selecting the best candidate from the final list. A sorting mechanism that sorts candidates based on prerequisite information allows for the inclusion of the Honey and Mumford model.

In contrast to the approach adopted in INSPIRE, APeLS developers provide a separate concept representation for each learning style of the Honey and Mumford model. A certain concept representation is chosen to be included into the final presentation based on the user's learning style. The user is allowed to experience other learning styles by selecting an alternative representations. Thus the learner is supposed to read the contents of one concept representation to fully understand the concept while in INSPIRE the learner has to read all concept representations shown in a certain order.

In fact, APeLS can deal with any learning style model, if the appropriate metadata is included in the learner and content model. APeLS facilitates adaptive navigation through the sequencing of candidates in the narrative.

AES-CS

AES-CS (Adaptive Educational System based on Cognitive Styles) provides adaptation to learners with the FD/FI learning styles in terms of navigational tools, adaptive presentation and adaptive navigation support.

The system provides FD learners with such navigational support tools as concept map and graphic path indicator. The graphic path indicator is dynamically created and presents the current, the previous and the next topic. The graphic path indicator appears at the bottom of each page and illustrates clearly the local neighbourhood of a topic. An "advance" organizer is provided for FD learners while FI learners are provided with "post" organizers.

AES-CS provides a learner control option for FI learners and a program control option for FD ones. In the first case, AES-CS provides a menu from which a learner can choose to proceed with the course in any order. In the program control option there is no menu, but the system guides the user through the learning material via adaptive navigation support. The system provides clear, explicit directions and the maximum amount of guidance to FD learners, whereas it provides minimal guidance and direction to FI learners. Moreover, it provides extensive feedback to FD learners, whereas it provides minimal feedback to FI learners.

Adaptive presentation in AES-CS is achieved through conditional text and the page variants representations. (Cited from [Triantafillou et al., 2002]): "With the conditional text technique, a page is divided into chunks. Each chunk of information is associated with a condition indicating which type of user should be presented with it. With page variants technique, two variants of the pages associated with a concept are prepared. Each variant of the page presents information in a different style according to FD/FI dimension."

Adaptive navigation support is provided through adaptive annotation and direct guidance. "The selection and the colour of hyperlinks are adapted to the individual student by taking into account information about the learner's knowledge state and the instructional strategy. Blue colour is used for 'recommended' and gray colour for 'not ready to be learned'. With the direct guidance, the system suggests to the student the next part of the learning material." "Student's prior knowledge is used by the system in order to provide him/her the most suitable sequence of knowledge units to learn and to work with. Furthermore, an annotation mechanism is used to show several levels of student's knowledge on each domain model concept. A colored checkmark is used to distinguish the state of student knowledge on any concept: a blue checkmark means 'know' (determined by the student through the student model), a red checkmark means 'learned' (the student has visited the pages which present the concept) and a green checkmark means 'well-learned' (the student has successfully completed a test)."

5.2.2 Methods for learning styles identification

Table 5.2 shows that most of the systems assess learning styles through the existing psychometric questionnaires. Some of the systems allow learners to change their learning style during interaction with the system (AES-CS, INSPIRE). Only a few systems provide some mechanisms for inferring learning styles (or preferences for a particular instruction that might correspond to certain learning styles).

In ACE (cited from [Specht and Opperman, 1998]) "in the introductory questionnaire, some information about the learners preferred learning style is captured and the system monitors which sequences of learning material are requested by a learner. For example, when a student often requests the introduction, looks for some examples, and then proceeds directly to the tests the system adapts to this learning strategy by presenting the links to the learning materials in this order. Strategies in which the learner shows a sufficient result in the final tests are rated as successful. Repeated occurrences raise the preference value of a style. Once a certain threshold is exceeded the learning strategy is taken as the default strategy for the learner."

MANIC uses a Naïve Bayes Classifier to reason about the learner's preferences in terms of explanations, examples and graphics by observing his/her interactions with the system.

In iWeaver the application of a Bayesian network is planned to predict and recommend media representations to the learner.

5.2.3 Discussion

From the review presented above we have seen that only few systems (ACE, APeLS) can provide certain types of adaptation independently of the learning styles model. In the rest, the learning styles are chosen and the corresponding instructional strategies are predefined by the designers of the systems. In our opinion this choice and definition should be done by the designers (authors) of the applications. We think that system developers should provide enough flexibility for the authors to define their own instructional strategies and to decide which strategies to apply for a particular application. Furthermore different authors can have different visions on the same instructional strategy — they may expect different kinds of adaptation for a user with a particular LS. This can be proved by the example, discussed above, on how INSPIRE and APeLS represent the learning styles of the Honey and Mumford model.

Another issue is that only a few systems (ACE, MANIC) provide mechanisms for inferring learners' preferences. In our opinion it would be useful to enrich the systems with mechanisms to unobtrusively detect learners preferences, which might indicate a certain LS. This could be done by observing how the user interacts with an application. In this way existing psychological questionnaires cannot be replaced completely, however some repetitive patterns in the user's browsing behavior can be observed. As a result, through the simplified and unobtrusive mechanisms, the dynamic adaptation of the system to identified user preferences can be provided.

In this dissertation we are trying to address these issues. Based on the previous reviews in the following section we identify the types of strategies that we foresee as needed in our system to provide adaptation to learning styles.

5.3 Identifying Types of Instructional (Meta-)Strategies

Our review of recommendations for LS application in Web-based environments and of the existing AHS providing adaptation to LS shows that it is possible to provide adaptation to the learner in terms of content adaptation, navigation paths or usage of multiple navigational tools. These adaptation types limit the possible response of the system to accommodate the different learning styles. The most frequently used elements of instructional strategies we have found in adaptive Web-based education literature are:

- *Selection of items* to accommodate different learner preferences or learning styles. For instance, the same information (or the same concept) can be presented in various ways by using alternative media types — audio, video, image, text, etc.. Depending on the learner's style a certain item (or group of items) may be included in the final presentation. As shown in table 5.1 the verbalizers may be presented with text and possibly spoken audio; whilst images, diagrams, graphs, charts or other items about the same concept can be given to the imagers. The selection process should be applied not only to media items, but also to other types of items.

  This type of strategy can be realized through the following low-level adaptation techniques as specified in Brusilovsky's taxonomy (see figure 2.1): adaptive multimedia presentation, adaptation of modality, altering fragments, stretchtext.

- *Ordering information or providing different navigation paths.* The order in which information items are processed can be based on individuals' learning styles. For instance, for active learners the information items should be presented in the following order: activity-example-theory-exercise; whereas for reflectors this order is different: example-theory-exercise-activity. The ordering process should be applied to other types of information items as well. Moreover the navigation paths can be adapted to support the global and analytic approach to learning, either showing first the overview concepts then going to details (breadth-first structure) or guiding the learner towards details of a concept before introducing the next one (depth-first structure).

  This type of strategy can be realized through the following low-level adaptation techniques: direct guidance, adaptive link sorting, adaptive link annotation, adaptive link hiding.

- *Providing learners with navigational support tools.* Depending on learner preferences, different learning tools can be provided. For example, as shown in table 5.1, FD learners can be provided with a concept map, graphic path indicator, advance organizer, etc., in order to help them organize the structure of the knowledge domain. Alternatively, FI learners might be provided with a control option showing a menu from which they can choose to proceed with the application in any order.

This type of strategy may require adaptation of the layout (not specified in Brusilovsky's taxonomy) and map adaptation.

To distinguish between the different types of strategies we need besides the previous list of elements of instructional strategies, a high-level classification based on their overall semantics. From the analysis of the literature, we observe that we can classify strategies according to their application range, as follows:

- *Instructional strategies* — define how the adaptation is performed. Namely the adaptation rules specified in the strategy are used to adjust the presentation to the learner with a particular learning style.

- *Instructional meta-strategies* — inference or monitoring strategies — are applied in order to infer the learner's preferences during his/her interaction with the system. These strategies cannot completely replace the existing psychological questionnaires for determining learning styles; however they can be used as a simplified, unobtrusive way to infer the learner preferences corresponding to these styles via their browsing behavior.

The first type of strategy is more frequently used, however as seen from the overview of the existing systems (section 5.2) the second type is still quite novel and requires some clarification. A meta-strategy can, for example, track the learner's preferences by observing his/her interactions with the system. It can track some repetitive patterns in the learner's behavior, like accessing particular types of information (if a choice of different types is available). It can observe that the user has a preference for textual information, which is typical for a learner with verbalizer style or, on the contrary, that the user has a preference for the pictorial representations (imagers or visualizers). It can also trace the navigational paths [Juvina, 2006]: browsing through the learning material in breadth-first order — typical for the learners with field-dependent or holist style — versus navigating in depth-first order, that might indicate a learner with analytic style. Meta-strategies of this type update some user model parameters which can be used later on for selecting a particular instructional strategy. These parameters can indicate what the system "thinks" the learner's preferences are. Meta-strategies could trace if the preferences specified by the learner when he begins working with the system stay the same or change. In case the learner's behavior is different from what was initially specified a strategy corresponding to another learning style might be suggested. Other examples of user model parameters which can be influenced by the actions specified in the meta-strategies are: level of difficulty of the material presented to the learner, link colors, etc. These actions occur when the learner accesses the concepts of an application. According to the type of adaptation provided we can refine the classification of adaptation strategies by analyzing the external (interactive) actions occurring, as presented in table 5.3:

Basic actions on items	Selection Showing the content of an item Showing a link to an item
Hierarchical actions on items	Actions on child items Actions on parent item
Actions on groups of items (e. g., siblings)	Ordering Performing "actions on items" on each group item
Actions on the *overall environment*	Changing the layout of the presentation

Table 5.3: Refined classification of actions in adaptive strategies

5.4 Summary

In this chapter we provided a review of the recommendations for application of learning styles in adaptive hypermedia from the psychological and computer science research. We also investigated how other researchers tried to incorporate learning styles into their AHS. These reviews provide us with an answer to research question 2 of this dissertation "Can we specify how an AHS should perform adaptation according to different learning styles?". The answer is "Yes". Both reviews showed us what can be the possible adaptive response of the system to a certain learning style (see table 5.1), moreover the second review presented the actual implementations (see table 5.2). In section 5.3, we classified the *instructional strategies* that define how adaptation to learning styles is performed as follows:

- *Selection of items* — this type of strategy can be implemented through the low-level adaptation techniques defined in Brusilovsky's taxonomy (see figure 2.1) such as adaptive multimedia presentation, adaptation of modality, altering fragments, stretchtext;
- *Ordering information or providing different navigation paths* — can be implemented through direct guidance, adaptive link sorting, adaptive link annotation, adaptive link hiding;
- *Providing learners with navigational support tools* — can be implemented through adaptation of the layout (not defined in Brusilovsky's taxonomy) and map adaptation.

By analyzing what the other researchers have done in order to implement it in our system as well we also identified a number of issues that we try to improve on in our approach:

- Most systems are bound to a certain learning style model and the designers of the systems define the response to the chosen learners styles. In our opinion this task should be left to the authors designing adaptive applications. The authors should be able to decide which learning styles to apply in a certain application and how a particular instructional strategy should look like. We argue that there is no single and "correct" way to represent a particular LS in AH. Different authors or psychologists may have a different view upon the representation of a given learning style. For example, they may want to use different adaptive presentation and adaptive navigation support techniques, different navigation tools, media types or concepts' representations, provide different order of items, etc..

- Most of the systems ask the learners to fill out long psychological questionnaires before proceeding with an application in order to identify their learning styles. In most cases there is no possibility of altering the questionnaire results during the further interaction with the system. We would suggest having an option for the learner to alter his/her learning style in order to try alternative strategies corresponding to different styles. We also propose having a simple option for inferring the learner's preferences if (s)he does not want to fill out the questionnaires.

  Because of these issues with the questionnaires we consider not only instructional strategies (corresponding to LS) but also the so-called *instructional meta-strategies* — LS inference or monitoring strategies.

Based on the provided classifications of strategies and actions performed through these strategies (see section 5.3) we incorporate learning styles in AHA! and present the implementation results in the following chapter.

Chapter 6

Incorporating Learning Styles in AHA!

In the previous chapter we identified the types of strategies that can be applied in adaptive hypermedia as a response to users' learning styles. We distinguish between *instructional strategies* such as selection of items, ordering information or providing different navigation paths, providing learners with navigational support tools and *instructional meta-strategies* or *monitoring strategies* that can be applied for inferring preferences for certain items, items order and navigation paths. To provide this strategy classification we looked through the chapter at the recommendations from the psychological literature as well as at the existing examples of AHS providing adaptation to learning styles.

The current chapter presents our approach to authoring and application of learning styles in the AHA! system. We follow the learning styles strategies classification to provide the corresponding adaptive behavior. While trying to incorporate in AHA! what other systems can do in terms of adaptation to learning styles we are also aiming at providing learning styles authoring support.

Section 6.1 describes the learning styles adaptation language we define for AHA!. It explains the choice of language format and the type of strategies that can be currently created with it. It also shows example strategies created in this language.

Section 6.2 explains how the strategies are translated into the AHA! domain/adaptation model. It shows how the strategies are applied by authors of AHA! applications and how they are visualized by end-users.

Section 6.3 provides the results of validation and evaluation of our approach. First, it refers back to AHS providing adaptation to learning styles discussed in the previous chapter and shows if and how AHA! implements the adaptation features of these systems. Second, it presents the results of evaluation of our system from the point of view of authoring ease and satisfaction with the resulting presentation which we performed with students of our university. Finally, it presents the results of validation of our system through interviews with and questionnaires for a number of psychologists with expertise in the field of learning styles. By presenting the actual implementation this chapter answers research question 3, 3a and 3b of this dissertation.

6.1 Learning Styles Language Definition

6.1.1 Choice of the language format

Merrill [Merrill, 2000] claims that "appropriate, consistent instructional strategies are determined first on the basis of the type of content to be taught or the goals of the instruction (the content-by-strategy interactions) and secondarily, learner style determines the value of the parameters that adjust or fine-tune these fundamental learning strategies (learning-style-by-strategy interactions)." Also "content-by-strategy interactions take precedence over learning-style-by-strategy interactions regardless of the instructional style or philosophy of the instructional situation."

Thus in order to be able to provide adaptation of the learning material to learning styles various representations of the domain model concepts should be available (e. g., different media items or items representing different aspects of the concepts) or various routes through the content should be possible. When the content is available the authors have to specify the adaptation rules for the domain model concepts defining their representation under various conditions or their order of presentation for different learning styles. Typically these rules become hidden in the content because the authors have to deal with separate instances. Another problem is that much repetitive work is required since the author has to define the same type of adaptive behavior for these concepts. Furthermore if an author decides to apply another instructional strategy to an application all the adaptation rules have to be rewritten.

In [Stash and De Bra, 2004] and [Stash et al., 2004] we presented our approach to authoring of learning styles using the high-level tool Graph Author that comes with AHA! 3.0. The tool allows for the definition and application of generic adaptation rules that can be connected to learning styles. For example, in [Stash and De Bra, 2004] we defined prerequisite relationships between concepts that differ for field-dependent and field-independent learners. Using these relationships can already save a lot of authoring work because the author does not need to specify the low-level adaptation rules for separate concepts. Instead, while saving the application in the Graph Author, the relationships will be automatically translated into low-level rules used by the AHA! engine. However the author would still have to draw all these relationships between separate concepts in the application graph in the Graph Author tool.

To be able to deal with these problems our proposal is to extract the adaptive response of the system from the content and to define the adaptation rules in a domain-independent high level form that can later on be applied to separate instances of various applications. By "high-level" form we mean that the language should deal with "generic concepts" and "generic rules" instead of instances. In [Stash et al., 2005] we called it a "learning styles adaptation language".

The strategies described in this language should contain a set of instructions that will inform the system how to adapt to a certain learning style or the rules for inferring the learner's preferences through his/her browsing behavior. Below we formulate the ***requirements*** for such a language.

6.1.1.1 Requirements for learning styles adaptation language

Two main requirements the e-Learning framework should satisfy are *interoperability* and *reusability* [e-learning Consortium, 2002]. (Note that AHA! is a general-purpose tool but in this part of the dissertation we are interested in the educational applications that can be created with it.)

Interoperability refers to the ability to take instructional components developed in one location with one set of tools or platform and use them in another location with a different set of tools or platform. Reusability refers to the flexibility to incorporate instructional components in multiple applications and contexts. In our research the instructional components represent the adaptive behavior (the actual dynamics) and not the static components such as the content of the domain model. Reuse of these items would be equivalent to exchanging not only the ingredients, but the recipes as well [Stash et al., 2005].

The language should be able to specify the types of strategies we identified in the previous chapter. However in the future we might want to be able to express different types of strategies that will require new additions. Thus another requirement is that the language should be *extensible* in the sense that it is easy to define the new elements.

Furthermore, the language should provide a *high level of semantics*, meaning the ability to define strategies containing constructs at a higher abstraction level than traditional IF-THEN-ELSE rules. Ideally the language should use or extend the emerging *Web standards*, which will enhance reusability and compatibility with current implementations [Stash et al., 2005].

To summarize, the main requirements for our "adaptation language" are as follows:
(a) interoperability,
(b) reusability,
(c) should use web standards,
(d) should be extensible,
(e) high level of semantics.

In order not to "reinvent" the wheel we looked in [Stash et al., 2005] at the existing standards and approaches to specifying the adaptive behavior independently of the application content. The focus of the paper was mainly on the reuse capabilities. We came to the conclusion that most standards deal with static material reuse. Below we provide a few alternatives for modeling and reuse of dynamics.

6.1.1.2 Review of standards and approaches for reuse of dynamics

Specifications such as IMS Simple Sequencing (IMS SS) and IMS Learning Design (IMS LD) deal with interoperability and reusability issues. Here we look at the features of these specifications.

IMS Simple Sequencing specification and the SCORM standard

One approach to specify the elements of the dynamics in a reusable way is to use the *IMS Simple Sequencing specification* (IMS SS) [ims, 2003b] or the *SCORM standard*[1] (which incorporates IMS SS specification). The IMS SS specification defines a method for representing the intended behavior of an authored learning experience such that any learning technology system can sequence discrete learning activities in a consistent way. A learning designer or content developer declares the relative order in which elements of content are to be presented to the learner and the conditions under which a piece of content is selected, delivered or skipped during presentation. The branching or flow of learning activities is based on the outcomes of a learner's interactions with content [ims, 2003b].

The IMS SS specification has been incorporated in the SCORM standard by ADL (Advanced Distributed Learning). SCORM is a collection of standards and specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reusability of Web-based learning content.

In [Romero et al., 2005] a conversion from SCORM into the AHA! system is shown. Only SCORM rules for sequencing as specified by IMS SS can be converted to AHA! prerequisite relationships (with adaptive link annotation). However, it does not provide other adaptive hypermedia techniques such as conditional inclusion of fragments that are possible in AHA!. Furthermore it does not touch the area of look and feel.

Thus IMS SS and consequently SCORM fail to provide all adaptive features adaptive hypermedia can offer.

IMS Learning Design specification

Another alternative is the use of *IMS Learning Design specification* (IMS LD) [ims, 2003a]. The IMS LD specification supports the use of a wide range of pedagogies in online learning. Rather than attempting to capture the specifics of many pedagogies, it does this by providing a generic and flexible language. This language is designed to enable many different pedagogies to be expressed [ims, 2003a].

The following adaptive techniques can be modelled using IMS LD — direct guidance, curriculum sequencing, show/hide links, link annotation, inclusion of pages [Berlanga et al., 2006]. However, like IMS SS and SCORM, IMS LD does not deal with content and layout adaptation.

IMS LD is being used in [Berlanga et al., 2006] for modeling predefined rules, techniques and learning elements. The rules are only at the level of assembly language of adaptation (according to the classification in [Cristea and Calvi, 2003]), i. e., IF-THEN rules, but are enriched with extra semantics. For this they use semantically labelled actions (such as show, hide, show-menu, sort-ascending, number-to-select, etc.). One problem with this approach is that it mixes user adaptation (such as some material being not recommendable for a user) with the actual presentation of this adaptation (hide it from user). This prob-

[1]http://www.adlnet.gov/scorm/index.cfm

lem is inherited from the strict adherence to the IMS-LD standard, which does not make this distinction. In the adaptive hypermedia literature [Brusilovsky, 2001], however, the presentation of an item which is undesirable can vary from hiding to color-code marking (e. g., "Red" is undesirable). This type of presentation depends on the degree of control the learner can have within the learning environment.

LAG language and grammar

Another approach to modeling and reuse of dynamics is provided by the LAG language [Cristea and Calvi, 2003] which has been briefly introduced in section 2.7 of chapter 2.

LAG [Cristea and Calvi, 2003] is a higher level language that allows for an increased level of semantics. We therefore take a closer look at its grammar (see figure 6.1).

```
PROG → DESCRIPTION VARS
        INITIALIZATION IMPLEMENTATION
DESCRIPTION → "// DESCRIPTION" COMMENT
VARS → ATTRIBUTE | (VARS)+ "," VARS
INITIALIZATION → "initialization("
                      STATEMENT ")"
IMPLEMENTATION →"implementation("STATEMENT ")"
STATEMENT → IFSTAT | WHILESTAT | FORSTAT |
  BREAKSTAT | GENSTAT | SPECSTAT | COMMENT |
  (STATEMENT) * STATEMENT | ACTION
IFSTAT → if CONDITION then (STATEMENT)
WHILESTAT → while CONDITION do
             (STATEMENT) [TARGETLABEL]
FORSTAT → for RANGE do (STATEMENT)
          [TARGETLABEL]
BREAKSTAT → break SOURCELABEL
GENSTAT → generalize((CONDITION)*)
SPECSTAT → specialize((CONDITION)*)
ACTION → ATTRIBUTE OP VALUE
COMMENT → "//""text" | (COMMENT)* COMMENT
CONDITION → enough((PREREQ)+, VALUE) | PREREQ
RANGE → "integer"
PREREQ → ATTRIBUTE COMPARE VALUE
LABEL → "text"
TARGETLABEL → "text"
SOURCELABEL → "text label a"
ATTRIBUTE → GENCONCEPT | SPECCONCEPT
GENCONCEPT → "CM type.concept.attr" |
            "CM type.concept.attr z"
SPECCONCEPT → "CM x.concept y.attr z"
OP → "=" | "+=" | "-=" | ".="
COMPARE → "==" | "<" | ">" | "in"
VALUE → "text"
```

Figure 6.1: The LAG grammar

The figure describes the components of an adaptive strategy "PROG". Each strategy has four main parts: description, variable declarations, initialization and implementation. The description is a comment for the human reader (the author who has to decide to apply this strategy or not). The variables are used to prevent overlaps and clashes if multiple strategies are applied on the same course: the use of these same variables should be informative about the possible clash. The initialization should set all the variables in use during the strategy, before the actual interaction of the strategy with the user (learner). It also establishes what learning items have to be shown to the user from the very beginning. The implementation part contains the actual user interaction, activity description. Initialization and implementation are built from statements. These building blocks are the basis of the LAG language. The adaptation language also allows assembly language statements, such as IF-THEN statements. However, it also contains more general programming statements, such as WHILE, FOR and BREAK statements and comments. The most specific statements are the SPECIALIZE and GENERALIZE statements, that allow the user to go down, or up the learning item hierarchy respectively — depending upon the fulfillment of certain conditions. These statements use the structure of the learning material, therefore have greater semantics for authors familiar with the learning material. The conditions are either prerequisites, or combinations of ENOUGH prerequisites. The value in the latter construct is a number, establishing how many of the prerequisites have to be fulfilled. In such a way more complex AND-OR combinations of conditions can be obtained. The details of the grammar have been simplified a little. However, it is important to remark that the ATTRIBUTES used in initializations, actions and comparisons can be of two main types: GENERAL or SPECIFIC. As in AHA! the specific attributes refer to an instance of the learning material, whereas the general attributes refer to materials of a given type. Therefore strategies can be written using only general attributes in order to be applied to any given set of learning materials, given the condition that these materials can be identified as being of a given type [Stash et al., 2005].

The adaptivity allowed is extremely flexible and the language is not domain dependent. User adaptation and presentation are kept separate. However one important drawback of the language is that it does not reflect current Web-standards.

6.1.1.3 Summary of the section

From the overview above we have seen that though IMS SS (and correspondingly SCORM) and IMS LD satisfy the reusability and interoperability requirements they allow only few features from what adaptive hypermedia can offer. LAG provides a very high level of semantics, it is very flexible and reusable, however it does not reflect any current web standards.

We therefore endeavor to invent our own adaptation language. In order to be completely AHA! compatible, we have chosen for an XML format. By this we mean that the domain/adaptation model and user models are described in AHA! in XML format (in addition to the possibility of their storage in mysql), also the concept templates and the templates for concept relationship types used by the Graph Author are described in XML. XML is a W3C standard (requirement (c) for the adaptation language), it is an extensible language (d) and is a cross-platform, software and hardware independent tool for representing and transmitting data (a). The language allows for inventing our own semantically enriched (e) elements. Reusability (b) can be achieved in the XML learning style adaptation language for AHA! by specifying each strategy as a separate XML file.

While defining our language we were aiming at providing the same high level of semantics as the LAG language does. We called the language LAG-XLS (read as "LAG-excels") since it instantiates the adaptation language layer of the LAG model with a specific goal aiming at learning styles strategies. In LAG-XLS we try to express the instructional strategies such as selection of items, ordering information or providing different navigation paths and the ability to express the monitoring strategies for item preferences and reading paths. Unfortunately interface adaptation cannot be provided by AHA! at the moment. Moreover the refined classification of actions is used as has been shown in table 5.3.

In the following section we describe the LAG-XLS language and its grammar.

6.1.2 An XML Learning Style Adaptation Language and Grammar: LAG-XLS

As discussed in the previous section in LAG-XLS we try to express the instructional strategies such as selection and ordering of concepts' representations, providing different navigation paths and the monitoring strategies for inferring preferences for types of information or reading order.

Our intention is to apply the strategies to the applications using the AHA! high-level authoring tool Graph Author. Therefore we will keep the syntax for LAG-XLS as close as possible to the syntax of the templates for concept types relationships used by the Graph Author. The adaptation strategies can be connected to concepts (domain model and other concepts such as concept "personal") through the attributes and values of these concepts. For the strategies definition we can use the child-parent and source-destination relationships as they are recognized by the Graph Author. For our approach we consider that the author defines concept representations as sub-concepts (defined as AHA! object concepts) of a parent (object or page) concept. The names of the attributes and their values indicate how these sub-concepts represent their parent concept. For instance, if the "media" attribute of the sub-concept is "audio" then it represents an audio version of its parent (see figure 6.2).

With these assumptions in mind we define the structure of the LS strategies as shown in figure 6.3 and provide a corresponding DTD of the LAG-XLS language in figure 6.4.

In both figures a number of qualifiers are used. They are used to indicate the number of time a certain element can appear in the strategy definition within its parent:

- ? — question mark — optional (zero or one time);
- * — asterisk — zero or more times;
- + — plus sign — one or more times.

The meaning of the DTD elements and attributes is explained in table 6.1.

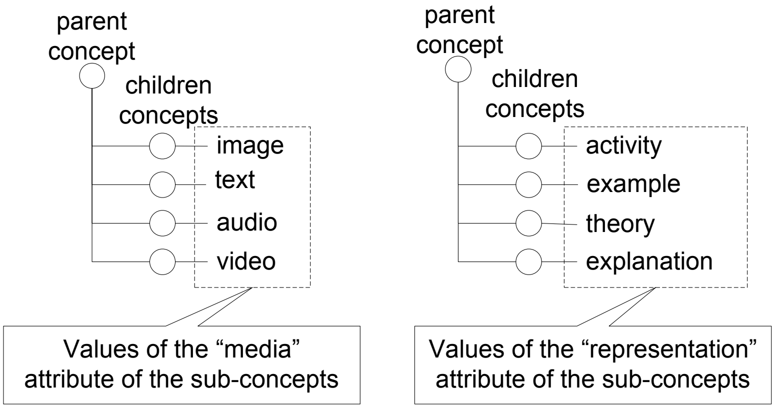


Figure 6.2: Defining concept representation in the AHA! domain model

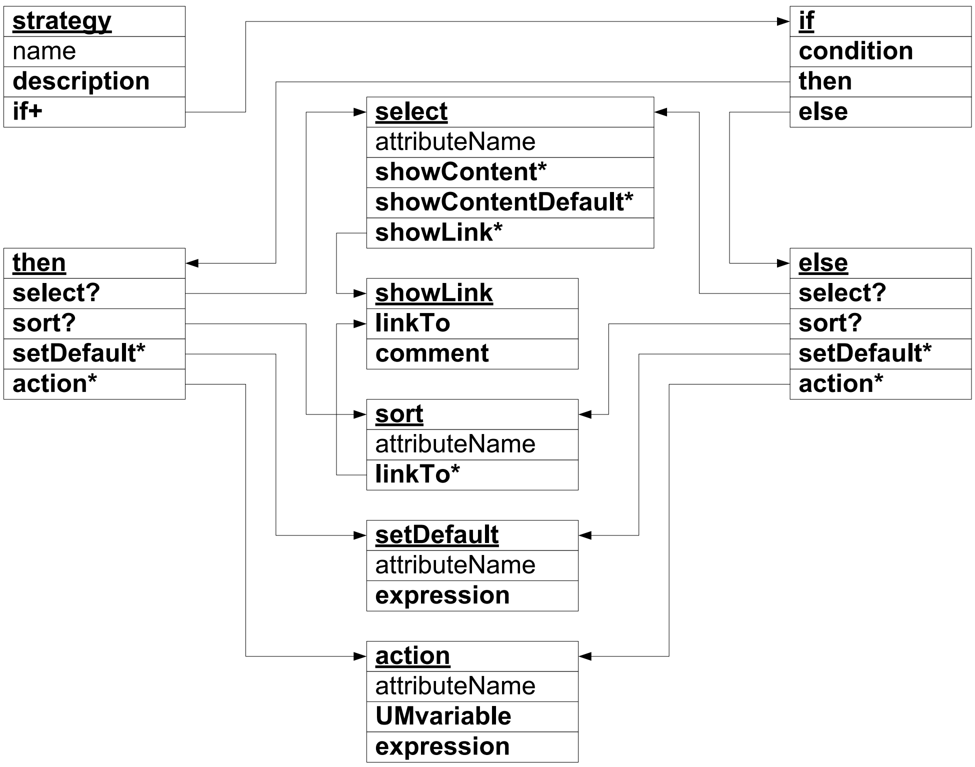


Figure 6.3: Learning styles strategies structure

Elements	**Description**
strategy	the root element of a file corresponding to a strategy
description	the strategy meaning, e. g., the corresponding learning styles model for which this strategy has been created
if	a statement to specify if-condition-then-else rules
condition	a Boolean expression which can contain some user-related information, e. g., information about the user's learning style
then	an element defining what happens when the condition is satisfied; this can be a *select* operation, then possibly a *sort* operation, then a series of *setDefault* operations, and finally a series of user model update *action*s
else	an element defining what happens when the condition is not satisfied
select	selecting a concept representation from a number of existing ones to be included in the final presentation; usage of a "select" element results in content adaptation
sort	sequencing different concept representations, reordering them from most to least relevant; usage of a "sort" element results in link adaptation
setDefault	setting the default values for the attributes of the domain model concepts through the value of the attribute *expression*
showContent	showing the content of the concept representation
showContentDefault	showing a default content specified by the author in case no other representation is found for a particular concept
showLink	showing a link to the concept representation
linkTo	specifies the type of concept representation link to which is provided
comment	the description of the link the system suggests to follow, e. g., "Follow this link to see more pictorial information"
action	specifies how the user model is updated when the user has accessed a concept in the domain model that has an attribute specified as *attributeName* within the "action" element
UMvariable	indicates which user model variable should be updated (which attribute of which concept), can be an absolute or relative update
expression	for a "setDefault" element it specifies the default value; for an "action" element it specifies the value used for user model update
Attributes	**Description**
name	the name of the strategy
attributeName	for "select" and "sort" elements this is the name of the attribute based on which values selection and sorting are provided; for "setDefault" — the name of the attribute which value is set to a certain default; for "action" — the name of the attribute the access to which causes actions to be performed

Table 6.1: Elements and attributes of the strategies

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT strategy (description, if+)>
<!ATTLIST strategy name CDATA #REQUIRED>
<!ELEMENT description (#PCDATA)>
<!ELEMENT if (condition, then, else)>
<!ELEMENT condition (#PCDATA)>
<!ELEMENT then (select?, sort?, setDefault*, action*)>
<!ELEMENT else (select?, sort?, setDefault*, action*)>
<!ELEMENT select (showContent*, showContentDefault*, showLink*)>
<!ELEMENT sort (linkTo*)>
<!ELEMENT setDefault (expression)>
<!ATTLIST setDefault attributeName CDATA #REQUIRED>
<!ATTLIST select attributeName CDATA #REQUIRED>
<!ATTLIST sort attributeName CDATA #REQUIRED>
<!ELEMENT showContent (#PCDATA)>
<!ELEMENT showContentDefault (#PCDATA)>
<!ELEMENT showLink (linkTo, comment)>
<!ELEMENT linkTo (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT action (UMvariable, expression)>
<!ATTLIST action attributeName CDATA #REQUIRED>
<!ELEMENT UMvariable (#PCDATA)>
<!ELEMENT expression (#PCDATA)>
```

Figure 6.4: LAG-XLS DTD

A quick comparison between LAG and LAG-XLS immediately shows that LAG-XLS is a much simpler language, without "for" and "while" constructs that could make the language Turing-complete. LAG-XLS concentrates on the adaptation that is essential for learning styles (see section 5.3): the *selection* of different representations of concepts (different media, different level of difficulty, different type of activity, etc.) and the *sorting* of concepts or representations of concepts, as needed to accommodate learning style pairs like FD/FI, activist/reflector, etc.. Section 5.3 also mentions the need for navigational support tools. The layout subsystem of AHA!, with different types of views offering menu-like navigation aids is clearly sufficient to offer the navigation support tools without requiring additional features in the LAG-XLS language for that.

To exemplify the use of the LAG-XLS, in the following sub-section we provide examples of some predefined strategies for AHA! applications.

6.1.3 Strategies Examples

In this section we provide a number of examples of the instructional (meta-)strategies as defined in LAG-XLS:

- instructional strategies for:
  - verbalizer versus imager style (section 6.1.3.1),
  - global versus analytic style (section 6.1.3.3),
  - activist versus reflector style (section 6.1.3.5);

- instructional meta-strategies for:
  - inferring preference for text or image (section 6.1.3.2),
  - inferring preference for navigation in breadth-first versus depth-first order (section 6.1.3.4).

6.1.3.1 Instructional strategy for verbalizer versus imager style

Here we present the definition of the instructional strategy for the verbalizer versus imager style pair. Recall that verbalizers prefer textual representations and imagers prefer images, representing the same concepts.

First, we have to decide how to store the information about the user's learning style. Learning style is a feature that is domain-independent but characteristic to the user himself/herself. We therefore do not wish to store this information in a concept that is artificially added to the concept structure of an AHA! application, but we add it to the application-independent concept "personal" that AHA! provides to store application-independent properties of the user. In AHA! we can add arbitrary attributes to concepts, including the "personal" concept. We have chosen to call the new attribute "VERBvsIM", and to make it an integer attribute, for which we will use values between 0 and 100. We interpret the values as follows:t when the value is less than 30 the user is most probably an imager, with the greatest confidence when this value is 0. When the value is greater that 70 the user is most probably a verbalizer, with the greatest confidence when the value is 100. When the value is in-between we consider that the user's style is not (yet) known. The value of the "VERBvsIM" attribute is initialized through the registration form when the learner starts working with the application. The strategy definition is shown in pseudo-code in figure 6.5.

The meaning of the strategy is that if the user is an imager (personal.VERBvsIM<30) then, for each concept that can be represented by different media types, an "image" representation is included in the presentation. If no "image" representation exists, then the default representation provided by the author is used. The author can also specify that links to other concept representations should be included. In the example a link to a textual representation is inserted.

On the other hand, if the user is a verbalizer (personal.VERBvsIM>70) then, a textual representation (or "default" if "text" representation does not exist) is included. Also a link to pictorial representation is provided.

If the user's style is not known (personal.VERBvsIM>29 && personal.VERBvsIM<71) then the material which is more or less suitable for both styles is presented (the "default" representation in the example). Also the links to pictorial and textual information are provided.

Linking resources (files) to concept representations is done by the author, so it is the author's responsibility to ensure that the textual representation corresponds to a text fragment and the pictoral representation to an image.

```
Strategy name: "VerbalizerVersusImager"
Description: Instructional strategy for verbalizer versus imager style
if: personal.VERBvsIM < 30
then: Selection process based on the "media" attribute
  showContent: image
  showContentDefault: default
  showLink:
    linkTo: text
    comment: <img src="/aha/AHAtutorialLS/images/text.gif" />
             more textual information
if: personal.VERBvsIM > 70
then: Selection process based on the "media" attribute
  showContent: text
  showContentDefault: default
  showLink:
    linkTo: image
    comment: <img src="/aha/AHAtutorialLS/images/image.gif" />
             more pictorial information
if: personal.VERBvsIM > 29 && personal.VERBvsIM < 71
then: Selection process based on the "media" attribute
  showContent: default
  showLink:
    linkTo: image
    comment: <img src="/aha/AHAtutorialLS/images/image.gif" />
             more pictorial information
  showLink:
    linkTo: text
    comment: <img src="/aha/AHAtutorialLS/images/text.gif" />
             more textual information
```

Figure 6.5: Instructional strategy for verbalizer versus imager style

```
Strategy name: "TextVersusImagePreference"
Description: Instructional meta-strategy for inferring preference
             for image or text
(1)
if:
(a) personal.traceTextvsImage &&
(b) personal.initialVERBvsIM > 29 && personal.initialVERBvsIM < 71 &&
(c) concept.media == "text" &&
(d) concept.visited == 0 &&
(e) parent.image == 0
then: User model update based on access to a concept with attribute "media"
    personal.VERBvsIM := personal.VERBvsIM+10
(2)
if: personal.traceTextvsImage &&
    personal.initialVERBvsIM > 29 && personal.initialVERBvsIM < 71 &&
    concept.media == "image" && concept.visited == 0 && parent.text == 0
then: User model update based on access to a concept with attribute "media"
    personal.VERBvsIM := personal.VERBvsIM-10
(3)
if: personal.initialVERBvsIM > 29 && personal.initialVERBvsIM < 71 &&
    personal.traceTextvsImage && personal.VERBvsIM < 30
then: User model update based on access to a concept with attribute "media"
    personal.mediaPreference: = "image"
    personal.traceTextvsImage = false
(4)
if: personal.initialVERBvsIM > 29 && personal.initialVERBvsIM < 71 &&
    personal.traceTextvsImage && personal.VERBvsIM > 70
then: User model update based on access to a concept with attribute "media"
    personal.mediaPrefence: = "text"
    personal.traceTextvsImage = false
```

Figure 6.6: Instructional meta-strategy for inferring preference for images or text

6.1.3.2 Instructional meta-strategy for inferring preference for text versus image

Figure 6.6 presents the pseudo-code of the monitoring strategy for inferring a user's preference for text or images. The meaning of the strategy is that if the system observes the user's repeated accesses, e. g., to textual items while pictorial items have not been visited, it increases the confidence that the user is a verbalizer, correpondingly decreasing the confidence that (s)he is an imager, and vice versa for repeated accesses to pictorial information.

The generic names used in the strategy description — "concept" and "parent" — will be translated into the actual concept names while applying the strategy to a particular application. The code contains a number of additional variables we invent for the current example.

For this meta-strategy we introduced some additional attributes for the concept "personal":

- traceTextvsImage — indicates whether the user wants the system to infer his/her preferences or not. For example, the user does not know what his/her learning style is and wants the system to trace it.

  If tracing is desired the value of "personal.traceTextvsImage" is set to "true".

- initialVERBvsIM — stores the initial value of the "VERBvsIM" attribute. The value of "VERBvsIM" changes during the user's interaction with the system while "initialVERBvsIM" stays the same. The values are compared by the system at each user's step (at each access to pictorial or textual information) to see what the user's initial preference was and whether a threshold (30 or 70 in the example) indicating a particular learning style has been reached.

- mediaPreference — indicates the media preference as identified by the system and will be used for displaying a message to the user about the inferred preference.

The values of these variables are initialized through the registration form. The user can thus opt-in or opt out of having the verbalizer/imager style traced by the system.

In order for the system to trace the media preference it needs to know whether concept representations that are being accessed are textual or pictorial. In or to do so the following attributes are added to all domain model concepts:

- "media" attribute of a concept with the String values "image" and "text";

- "image" and "text" attributes of a "parent" concept which have integer values indicating whether a corresponding pictorial or textual representation has been visited.

The first "if" statement of the strategy indicates that when
(a) the user wants the system to trace his/her preference (personal.traceTextvsImage) and
(b) his/her learning style is not known
(personal.initialVERBvsIM>29 && personal.initialVERBvsIM<71) and
(c) the user has accessed the textual item (concept.media == "text")
(d) which has not been visited before (concept.visited == 0)
(e) while the image representation has not been seen (parent.image == 0)
the system increases the confidence that the user is a verbalizer, indicated by the expression "personal.VERBvsIM+10". This action correspondingly decreases the confidence that the user is an imager. Increase of "personal.VERBvsIM" attribute by 10 has been chosen based on a sample application we have used. But in general this can be any arbitrary value.

Correspondingly, as indicated by the second "if" statement, repeated accesses to pictorial representations increase the confidence of the system that the user is an imager, indicated by "personal.VERBvsIM-10".

The third "if" statement indicates that when the user's learning style is initially not known and (s)he wants the system to trace his/her preference and the value of "VERBvsIM" has reached the threshold of 30 the system infers that the user has a preference for pictorial material. It sets the value of "personal.mediaPrefence" to "image" and "personal.traceTextvsImage" to "false". The tracing will stop and an instructional strategy for the imager will be suggested to the user.

Similarly, the fourth "if" statement indicates that the system will trace the user's behavior until a threshold of "70" is reached and after that suggest him/her a strategy for the verbalizer.

If the learner is not satisfied with an instructional strategy (s)he can always inspect his/her user model and make necessary corrections. (S)he might still let the system trace his/her preferences even if (s)he explicitly specified what his/her learning style is. For this more adaptation rules should be specified in the strategy in addition to the shown ones.

6.1.3.3 Instructional strategy for global versus analytic style

We also predefined an instructional strategy for global versus analytic learners. Through this strategy we provide a breadth-first versus depth-first navigatino structure for the application it applies to. The pseudo-code of the strategy is presented in figure 6.7.

Similar to the previous example to indicate global versus analytic learning style we add a new integer attribute "GLvsAN" to the concept "personal" for which we use values between 0 and 100.

The meaning of the strategy is that if the learner has a global style(personal.GLvsAN > 70) an application is presented using a breadth-first navigation structure; if the learner has analytic style (personal.GLvsAN < 30) the same application is presented using a depth-first navigation structure, and if the value is in-between, the user is provided with a freedom of navigation. (By observing how the user actually uses this freedom the preference for breadth-first or depth-first can then be deduced by a meta-strategy.)

For the implementation of this strategy we also need some auxiliary "variables". Because these do not belong to any concept of the application we have created them as attributes of the concept "personal". Note however that the variables do not really represent a property of the user, and their use can become problematic if several applications are used simultaneously by the same user on the same AHA! server.

- numberConceptsLevel{i} — number of concepts at a level "i" in the hierarchy;
- numberConceptsVisitedLevel{i} — number of visited concepts at level "i" in the hierarchy;
- upToLevel — level up to which the breadth-first or depth-first order is provided.

To each domain model concept we add the following attributes in order to be able to determine the position of concepts in the concept hierarchy:

- level — indicates the concept's level in the domain model hierarchy, e. g., the root concept has level 0, its children have level 1, their children have level 2 and so on;
- parent.firstChild — first child of a parent concept;

```
Strategy name: "GlobalVersusAnalytic"
Description: Instructional strategy for global versus analytic style
(1)
if: parent.firstChild == concept.name
then: set default of the concept.suitability to
(a) (personal.GLvsAN > 29 && personal.GLvsAN < 71) ||
(b) (personal.GLvsAN < 30 &&
(b1)((concept.level < personal.upToLevel && parent.visited > 0) ||
(b2)(concept.level > personal.upToLevel-1 &&
     ancestorAt{personal.upToLevel-1}.visited > 0))
(c) (personal.GLvsAN > 70 &&
(c1)(concept.level < personal.upToLevel &&
    personal.numberConceptsLevel{concept.level-1} ==
    personal.numberConceptsVisitedLevel{concept.level-1}) ||
(c2)((concept.level > personal.upToLevel-1) &&
    personal.numberConceptsLevel{personal.upToLevel-1} ==
    personal.numberConceptsVisitedLevel{personal.upToLevel-1}))
(2)
if: parent.firstChild != concept.name
then: set default of the concept.suitability to
(d) (personal.GLvsAN > 29 && personal.GLvsAN < 71) ||
(e) (personal.GLvsAN < 30 &&
(e1)(concept.level < personal.upToLevel &&
    (previousSibling.visited > 0 && previousSibling.knowledge > 75) ||
(e2)(concept.level > personal.upToLevel-1 &&
     ancestorAt{personal.upToLevel-1}.visited > 0)) ||
(f) (personal.GLvsAN > 70 &&
(f1)(concept.level < personal.upToLevel &&
    previousSibling.visited > 0) ||
(f2)(concept.level > personal.upToLevel-1 &&
    personal.numberConceptsLevel{concept.level-1} ==
    personal.numberConceptsVisitedLevel{concept.level-1}))
(3)
if: concept.suitability && concept.visited == 0 &&
    (concept.concepttype == "page" || concept.concepttype == "node")
then: User model update based on access to a concept with attribute "level"
personal.numberConceptsVisitedLevel{concept.level}++
```

Figure 6.7: Instructional strategy for global versus analytic style

- previousSibling — previous sibling of a concept;
- ancestorAt{i} — ancestor of a concept at level "i".

In order to provide breadth-first or depth-first paths we use the "setDefault" element of the LAG-XLS DTD to set the default value for the suitability attribute of all domain model concepts. We distinguish a number of cases:

1. whether a concept is a first child of its parent or not;
2. whether the concept is above the "personal.upToLevel" level in the domain model hierarchy or not;
3. what the user's learning style is: the user's style is either not known or the user is assumed to have analytic or global style.

Below we explain the effect of the strategy. We also provide the figures visualizing the discussed cases corresponding to different locations of the concepts in the domain model hierarchy (see figures 6.8 and 6.9). On the figures the discussed cases are indicated as b1, b2, etc. and the filled in circles mean that corresponding concepts have been visited.

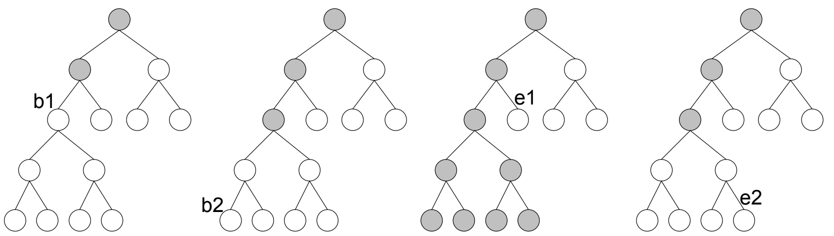


Figure 6.8: Desirability of concepts for learner with analytic style

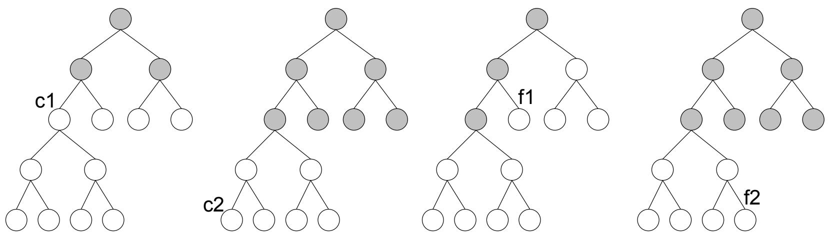


Figure 6.9: Desirability of concepts for learner with global style

The first "if" statement of the strategy specifies which default value for a concept which is the first child of its parent (parent.firstChild == concept.name) is being set. This concept is desirable if:

(a) the user's style is not known;
(b) the user has analytic style and
(b1) the concept is at a higher level in the hierarchy than the "up to" level (concept.level < personal.upToLevel) and its parent has been visited (parent.visited > 0);
(b2) the concept is at or below the "up to" level in the hierarchy (concept.level > personal.upToLevel-1) and the ancestor concept of the current concept at the level above the "up to" level has been visited (ancestorAt{personal.upToLevel-1}.visited > 0).
(c) the user has global style and
(c1) the concept is at a higher level in the hierarchy than the "up to" level and all concepts at the level above that concept level are visited (personal.numberConceptsLevel{concept.level-1} == personal.numberConceptsVisitedLevel{concept.level-1});
(c2) the concept is at or below the "up to" level and all concepts at the level above the "up to" level are visited (personal.numberConceptsLevel{personal.upToLevel-1} == personal.numberConceptsVisitedLevel{personal.upToLevel-1}).

The second "if" statement specifies which default value for a concept which is not the first child of its parent (parent.firstChild != concept.name) is being set.

Similar to the previous case this concept is desirable if:
(d) the user's style is not known;
(e) the user has analytic style and
(e1) the concept is at a higher level in the hierarchy than the "up to" level, its previous sibling has been visited and all concepts in the subtree of the previous sibling have been read (previousSibling.visited > 0 && previousSibling.knowledge > 75).
For concepts defined using the page concept template this expression would mean that only a previous sibling has to be visited and read but it will not check whether the concepts in its subtree have been read. Therefore we had to define an extra concept template, the so-called "node concept", which has the same attributes as a page concept but it only specifies the visited attribute update. We define all concepts in the domain model that are not leaves in the hierarchy as the node concepts. The knowledge value of these concepts will be the knowledge accumulated through the knowledge propagation from the leaf concepts (defined as page concepts). Note that this propagated knowledge will be high enough only if the accessed leaf concepts were desirable.
(e2) the concept is at or below the "up to" level in the hierarchy and the predecessor concept of the current concept at the level above the "up to" level has been visited.
(f) the user has global style and
(f1) the concept is at a higher level in the hierarchy than the "up to" level and the previous sibling has been visited (previousSibling.visited > 0);
(f2) the concept is at or below the "up to" level in the hierarchy and all concepts at the level above the "up to" level are visited (personal.numberConceptsLevelpersonal.upToLevel-1 == personal.numberConceptsVisitedLevelpersonal.upToLevel-1).

The third "if" statement indicates that when the user has accessed a desirable concept (concept.suitability) which has not been visited before and this concept is of a "page" or "node" type (concept.concepttype == "page" || concept.concepttype == "node") a number of concepts visited at the level of the currently accessed concept is increased by 1 (we do not count the situation when the user encounters "object" concepts on the accessed pages).

6.1.3.4 Instructional meta-strategy for inferring preference for navigation in breadth-first versus depth-first order

Figure 6.10 presents the pseudo-code of the strategy for inferring preference for navigation in breadth-first versus depth-first order. The idea of the strategy is that we compare the number of concepts visited at different levels. For example, if we see that the user visited more concepts at a higher level we increase the confidence that the user prefers a breadth-first structure. Otherwise we increase the confidence that (s)he prefers a depth-first structure. After a number of steps, when a threshold specified by the author is reached (30 or 70 in our example) we consider the attribute value to be meaningful for indicating the user's preference.

For this meta-strategy we add two new attributes to the concept "personal":

- traceBFvsDF — indicates whether the user wants the system to trace his/her preference for navigation;
- level — indicates the current position of the user in the domain model hierarchy which we compare with the "level" attribute of the accessed concepts and
- navigationPreference — indicates the reading order preference as identified by the system and will be used for displaying a message to the user about the inferred preference.

Note again that (1) by setting the traceBFvsDF value through the application's registration form the user can opt-in or opt-out for having this learning style traced, and that (2) by again using the "personal" concept for an auxiliary variable ("level") the user should not be using more than one application on the same AHA! server at the same time.

The condition of the first "if" statement indicates the situation when
(a) the user specified through the registration form that (s)he wants the system to trace his/her preference for navigation (personal.traceBFvsDF was set to "true"),
(b) (s)he has not visited the accessed concept before (concept.visited == 0),
(c) this accessed concept is of the "node" or "page" type and
(d) the concept is at the same level in the domain model hierarchy as the previously visited by the user (concept.level == personal.level) and
(e) the concept was not the "root" concept (personal.level != 0).

As the user stays at the same level in the hierarchy the system increases his/her preference for navigation in breadth-first order that corresponds to a greater assurance that the user has a global style (personal.GLvsAN+5).

```
Strategy name: "BFVersusDFPreference"
Description: Instructional meta-strategy for inferring preference
             for breadth-first versus depth-first order of navigation
(1)
if:
(a) personal.traceBFvsDF &&
(b) concept.visited == 0 &&
(c)(concept.concepttype == "node" || concept.concepttype == "page") &&
(d) concept.level == personal.level &&
(e) personal.level != 0
then: User model update based on access to a concept with attribute "level"
     personal.GLvsAN := personal.GLvsAN + 5
(2)
if: personal.traceBFvsDF && concept.visited == 0 &&
    (concept.concepttype == "node" || (concept.concepttype == "page") &&
    concept.level > personal.level &&
    personal.numberConceptsVisitedLevel{personal.level} <
    personal.numberConceptsLevel{personal.level}
    && personal.level != 0
then: User model update based on access to a concept with attribute "level"
     personal.GLvsAN := personal.GLvsAN - 5
(3)
if: concept.visited == 0 &&
    (concept.concepttype == "node" || concept.concepttype == "page")
then: User model update based on access to a concept with attribute "level"
     personal.level := concept.level
     personal.numberConceptsVisitedLevel{concept.level}++
(4)
if: (personal.initialGLvsAN > 29 && personal.initialGLvsAN < 70 &&
     personal.traceBFvsDF && personal.GLvsAN < 30)
then: User model update based on access to a concept with attribute "level"
     personal.navigationPreference: = "depth-first"
     personal.traceBFvsDF := false
(5)
if: (personal.initialGLvsAN > 29 && personal.initialGLvsAN < 70 &&
     personal.traceBFvsDF && personal.GLvsAN > 70)
then: User model update based on access to a concept with attribute "level"
     personal.navigationPreference: = "breadth-first"
     personal.traceBFvsDF := false
```

Figure 6.10: Instructional meta-strategy for inferring preference for breadth-first versus depth-first order of navigation

In contrast with the condition of the first "if" statement the condition of the second one indicates the situation when the user has accessed a concept at a deeper level in the hierarchy than the previously visited one (concept.level > personal.level) while not all the concepts at the level where the user has been before have been visited
(personal.numberConceptsVisitedLevel{personal.level} <
personal.numberConceptsLevel{personal.level}).

As the user goes deeper in the hierarchy the system increases his/her preference for navigation in depth-first order that corresponds to a greater assurance that the user has a analytic style (personal.GLvsAN-5).

The third "if" statement indicates that when the user has accessed a concept that has not been visited previously and this concept is of a "node" or "page" type the current location of the user in the hierarchy is set to the level of currently accessed concept (personal.level := concept.level) and the number of visited concepts at that level is increased by 1 (personal.numberConceptsVisitedLevel{concept.level}++).

The fourth "if" statement indicates that when the user's learning style is initially not known (personal.initialGLvsAN > 29 && personal.initialGLvsAN < 71) and (s)he wants the system to trace his/her preference and the value of "GLvsAN" has reached the threshold of "30" the system infers that the user has preference for depth-first navigation. It sets the value of "personal.navigationPrefence" to "depth-first" and "personal.traceBFvsDF" to "false". The tracing will stop and an instructional strategy for the analytic style will be suggested to the user.

Similarly, the fifth "if" statement indicates that the system will trace the user's behavior until a threshold of "70" is reached and after that suggest him/her a strategy for the global style.

6.1.3.5 Instructional strategy for activist versus reflector style

In the same way as in the previous examples, for the activist versus reflector strategy we add a new attribute, "HoneyMumfordLM" to the concept "personal". It is a string and has values "Activist" or "Reflector", indicating activists and reflectors respectively. The pseudo-code of the strategy is presented in figure 6.11.

The meaning of the strategy is that the links to children of a concept that represent it by activity, example, explanation and theory will be annotated and reordered depending on the user's style. For an activist (personal.HoneyMumfordLM == "Activist") first a link to an activity will be shown as "good" while links to other representations will be shown as "bad". When the user accesses an activity item the link to it turns to neutral and a link to the next item in the strategy list — "example" — becomes "good" and so on. Correspondingly for a reflector (personal.HoneyMumfordLM == "Reflector") first the link to an example will be shown as "good" and links to other representations as bad, etc..

Similarly, the strategy can define adaptation for pragmatists and theorists. We can also define a strategy for inferring the activist versus reflector style. We omit this because this can be done in a way that is very similar to the meta-strategy for inferring text versus image preference.

```
Strategy name: "ActivistVersusReflector"
Description: Instructional strategy for activist versus reflector style
if: personal.HoneyMumfordLM == "Activist"
then: Sort based on the "representation" attribute
      linkTo: activity
      linkTo: example
      linkTo: explanation
      linkTo: theory
if: personal.HoneyMumfordLM == "Reflector"
then: Sort based on the "representation" attribute
      linkTo: example
      linkTo: explanation
      linkTo: theory
      linkTo: activity
```

Figure 6.11: Instructional strategy for activist versus reflector style

6.1.3.6 Summary of the sub-section

In this sub-section we presented examples of adaptation strategies that can be created with LAG-XLS. The strategies can be reused by different authors of AHA! applications. The language is very flexible in the sense that for their own applications the authors might create their own visions of the predefined (meta-)strategies. They might use different attributes of different types indicating the user's styles (instead of those ones we specified in the examples), specify a different range of values for the attributes and different kinds of adaptation allowed by the DTD of LAG-XLS. They might define a different set of actions for inferring the learner's preferences, set different thresholds and different values for user model updates depending on their domain model content. The authors may consider different gradations of learning styles for providing adaptation. For example, instead of bipolar gradation for the verbalizer versus imager style which we use in our example, the authors might want to make a distinction between strong and mild preferences for both styles. The language allows to specify generic concepts and generic adaptation rules, however it can also deal with their instances.

In the following section we show how the learning styles strategies are translated to AHA! from the implementation point of view, how they are applied to AHA! applications from the authoring point of view and how they are visualized in the AHA! applications from the end-users point of view. The observant reader may have noticed already that whereas most of the translation seems straightforward the "sort" operation needed in the activist/reflector strategy requires an adaptation technique of which we indicated in chapter 3 that AHA! does not support it.

6.2 Applying LAG-XLS to AHA!

6.2.1 Adaptive strategies translation in AHA!: implementation issues

In this sub-section we will explain how the strategies described in the previous section are converted for the AHA! delivery engine.

The strategies are applied to the application through the high-level Graph Author tool. This tool translates the high-level constructs it recognizes into AHA! low-level (assembly) adaptation rules. Graph Author that comes with AHA! 3.0 recognizes such generic concepts as “parent” and “concept” which we use in the strategies definitions. However for global versus analytic strategy we need other types of generic concepts such as “previousSibling”, “ancestorAt{i}”, “parent.firstChild”. Therefore the Graph Author had to be extended with the possibility of recognizing and translating these constructs into AHA!.

The result of the strategies application can be as follows:

- extra attributes can be added to parent concepts (e. g., through verbalizer versus imager and activist versus reflector strategy);
- extra concepts and pages can be generated (for verbalizer versus imager strategy);
- the default values for the suitability attributes of the concepts can be set (global versus analytic and activist versus reflector);
- extra actions updating the user model can be added to the concepts (text versus image preference and breadth-first versus depth-first preference), as needed by the meta-strategies.

Let us have a look at how the strategy for verbalizer versus imager (as shown in figure 6.5) is translated to AHA!. The strategy has to be applied to all AHA! concepts in a given application which have sub-concepts with an attribute “media”. The application of the strategy should have as effect the display of the content of the appropriate sub-concept, depending on the value of the attribute (“image”, “default” or “text”), or only a link to inappropriate sub-concepts.

The strategy influences the structure of the parent concepts (see the simplified syntax in figure 6.12). A parent concept gets extra attributes “image” and “text” of integer type indicating whether the corresponding representation has been visited. Also the “showability” attribute of a parent concept is affected, which determines which fragment to show when a parent is accessed. Furthermore extra pages representing those fragments are generated (below we will show how).

The selection of the appropriate representation of a concept is performed using *adaptive resource selection*. Here we benefit from AHA!’s design that uses this single mechanism for both *adaptive link destinations* and the *conditional inclusion of objects*. The parent concept can be defined as an AHA! “page” or “fragment” concept. When the parent is a

```
Concept name: conceptname
Attribute name: access
Attribute name: image
Attribute name: text
Action
if: personal.VERBvsIM > 29 && personal.VERBvsIM < 71
then: conceptname.showability := 1
if: personal.VERBvsIM < 30
then: conceptname.showability := 2
if: personal.VERBvsIM > 70
then: conceptname.showability : = 3
Attribute name: showability
Casegroup
Casevalue:
value: 1; returnfragment: generatedfile4.xhtml
value: 2; returnfragment: generatedfile2.xhtml
value: 3; returnfragment: generatedfile5.xhtml
```

Figure 6.12: Example part of the generated structure for the AHA! concept

```
<span><object name="objectImage" type="aha/text" /><br />
<a href="generated1.xhtml">
You can also see the textual information about the same concept
</a></span>
```

Figure 6.13: Generated parent concept representation

page, selecting the appropriate representation becomes a case of adaptive link destinations. This is the case when the laerner follows a link to a parent concept. Depending on the (VERBvsIM attribute of the concept "personal" in the) user model the same link will lead to a different resource, showing a textual representation or an image.

When the concept is a fragment concept it is accessed by having a conditional object somewhere on a page. The adaptive resource selection performed to determine the resource to be included thus implements the *conditional inclusion of objects* technique. The resource will be a textual representation of the concept or an image.

Assume for instance that the parent concept is an object. Figure 6.12 then shows how it will be represented by generatedfile2.xhtml to a user with the imager style. The contents of this file are presented in figure 6.13.

The representation for an imager includes an image representation through an object tag since it is defined as an AHA! object concept. A link to a textual representation points to another generated file — generatedfile1.xhtml. The contents of this file are shown in figure 6.14.

```
<!DOCTYPE html SYSTEM "/aha/AHAstandard/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><body>
<object name="objectText" type="aha/text" />
</body></html>
```

Figure 6.14: Generated viewable representation of an object concept

This extra file is needed because the resources associated with the object concepts (e. g., resource associated with "objectText") can only be seen if included into pages. The goal of generated1.xhtml file is to add a header wrapper to this object concept to represent its viewable version.

Similarly, the resources representing the parent concept under other conditions (e. g., when the user is a verbalizer or his/her style is not known) will be generated (generated4.xhtml and generatedfile5.xhtml respectively).

Application of other strategies such as the global versus analytic style or activist versus reflector sets the default value for the suitability attributes of the concepts. For example, as a result of activist versus reflector strategy application the following expression is set for activity items in the application: personal.HoneyMumfordLM == "Activist" || (parent.theory > 0 && personal.HoneyMumfordLM == "Reflector") This means that an activity item becomes desirable if the learner is an activist or if the learner is a reflector and the theory item has been visited.

Unfortunately, the translation of the "sort" operation of LAG-XLS used in the definition of the strategy for activist versus reflector cannot be implemented in AHA! in an equally straightforward way as we were able to do for other operations. This is because of the fact that link sorting as required by the strategy is not automatically provided by AHA! (see figure 3.2), at least not in the main view — the view where the application content is displayed — or the standard views for navigation aids, like *TreeView* and *StaticTreeView*.

However, the layout possibilities of AHA! allow for creating additional views. Creating such a view requires some knowledge of Java programming language[2]. Java code for a new view should be written and then placed in the layout (code) part of the AHA! server and then a reference to this view should be added in the layout definition. (This reference is purely by name. No steps are needed to "register" a new view with the server in any way.)

For providing the link sorting option we created a view that shows a path from the root concept in the domain model to the currently accessed concept together with its siblings and children. The children of an accessed concept and the siblings are sorted based on their desirability. Since the various representations of a concept are defined as the children of that concept links to them will be sorted in this view. We will talk more about this in section 6.2.3.

[2]http://en.wikipedia.org/wiki/Java_programming_language

Strategies such as text versus image preference and breadth-first versus depth-first strategies add extra actions to all concepts that have an attribute specified within the "action" element of the strategy, e. g., "media" or "level".

In the following sub-section we present an authoring environment for application of learning styles strategies in AHA!.

6.2.2 Application of learning styles strategies in AHA!: authoring environment

There are a few considerations that the authors should keep in mind while designing an adaptive application providing support for learning styles using AHA!:

1. As has been discussed in section 6.1.2 the concept representations should be defined as sub-concepts of a concept they represent.

2. These concept representations should be defined as AHA! object concepts.

3. An object concept corresponding to a certain representation should have an attribute indicating how it represents its parent concept, e. g., "media" attribute with value "audio" or "representation" attribute with value "theory". This attribute will be used by the strategies applied to an application.

4. If an author wants to inform the user that a certain preference has been identified extra code should be added to the application content pages. For example, the following code can be added before the page title in the XHTML pages of an application that uses verbalizer versus imager and text versus image preference strategy:

   ```
   <if expr="personal.mediaPreference == ‘image’ ">
   <block>
   <font color="#FF0000">
   <b>Preference for pictorial information has been identified</b>
   </font><br />
   (<a href="ViewGet/applicationname/xhtml/LS-form.frm">
   Check your user model settings</a>)
   </block>
   </if>
   ```

   The code uses an "if" statement to display a message about the inferred preference based on the value of "personal.mediaPreference" attribute that is set through the text versus image preference strategy ("image" in the current example). The user has to follow a link to the special AHA! form to check his/het user model settings. Through this form the user can ask the system not show a message about the inferred preference again.

Further the sub-section presents an authoring tool for creating/editing the learning styles strategies and an extended version of the Graph Author tool allowing for strategies application.

6.2.2.1 Authoring tool for designing learning styles strategies

In section 6.1.2 we defined the LAG-XLS language and in 6.1.3 we showed some pseudo-code for the predefined strategies. Skilled authors having knowledge of XML can create and edit the XML files corresponding to learning styles strategies manually. For less skilled authors we provide an authoring tool that hides the XML syntax of the strategies files. It allows authors to add new statements by selecting them from a list of existing ones (as specified in LAG-XLS DTD). The statements can be added, edited and removed.

Figures 6.15 and 6.16 present screen shots for entering the strategy for verbalizer versus imager and activist versus reflector style respectively.

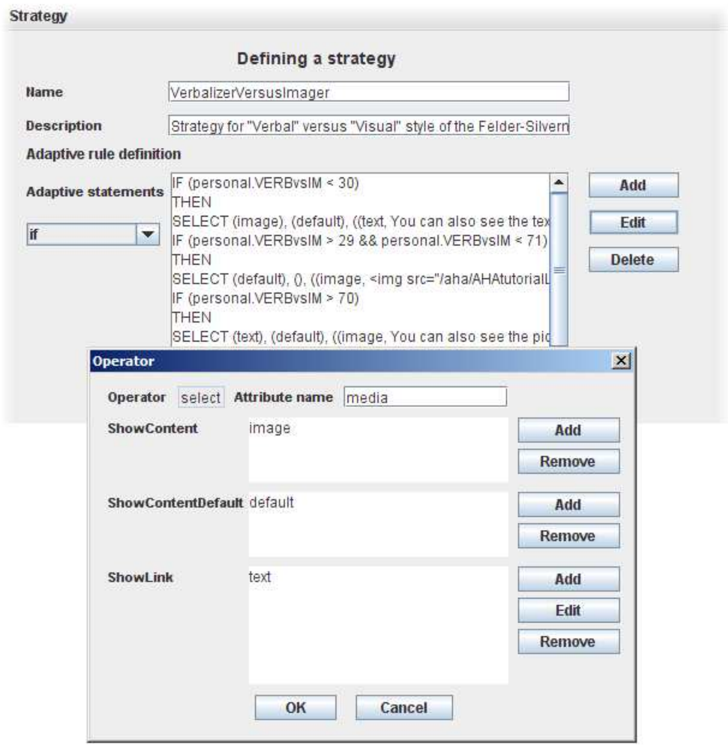


Figure 6.15: Tool for strategies creation

The created strategies files will be put into the author's directory, available only for himself/herself. A set of "standard" strategies which can be reused by all authors is available and will be extended. If authors want to let others use some of their strategies, they would have to add them to the list of standard strategies via the AHA! administrator.

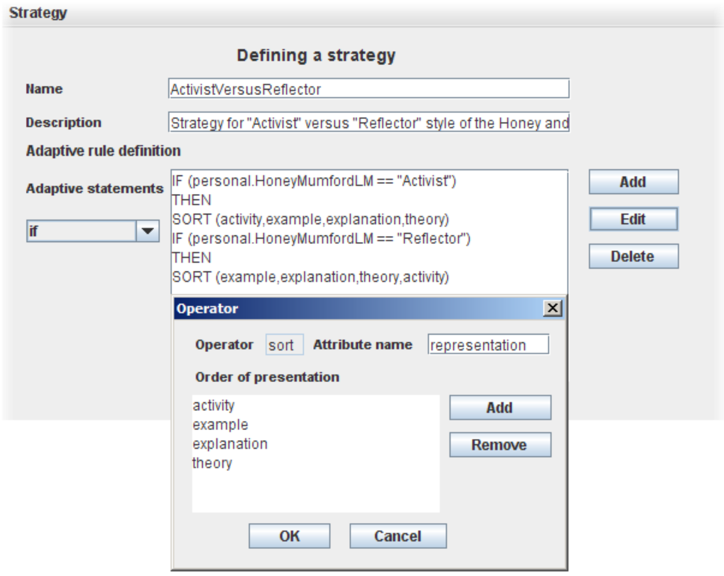


Figure 6.16: Tool for strategies creation

6.2.2.2 Strategies application in the Graph Author tool

After the strategies have been created manually or using the authoring tool the author has to apply these strategies in the Graph Author tool. A new option has been added to the tool (figure 6.17), allowing an author to choose which strategies to apply to a particular course, and in which order. Order can be very important in case of application of several strategies. For example, if there is a conflict between the applied strategies, such as one strategy specifies that a concept should be represented with an image while another strategy specifies that under the same conditions this concept should be represented as text, the strategy which appears higher in the list of applied strategies will take precedence and a representation specified in this strategy will be chosen.

During saving, the AHA! concept relationships graph is translated into AHA! low-level (assembly) adaptation rules. The details of this translation have been already discussed in section 6.2.1.

After that the strategies application can be visualized by end-users in the browser.

6.2.3 Strategies visualization in the browser

We have created two example applications:

- “AHAtutorialLS”: a tutorial about AHA! providing support for learning styles. Two instructional strategies — verbalizer versus imager and global versus analytic — and two instructional meta-strategies — text versus image and breadth-first versus depth-first preference — can be applied to it.

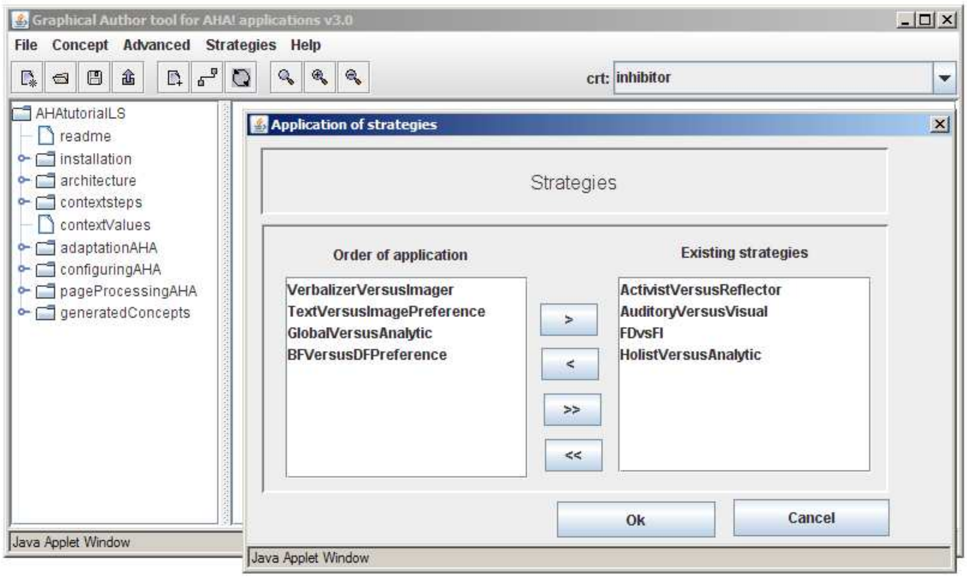


Figure 6.17: Adding strategies in the Graph Author tools

- a smaller example called "WritingApplets" which demonstrates how the application of activist versus reflector instructional strategy works.

Before accessing each of the applications the user has to fill out the registration form (see figure 6.18). Through this form the user sets his/her learning styles preferences and an indication for the system whether (s)he wants to trace them or not.

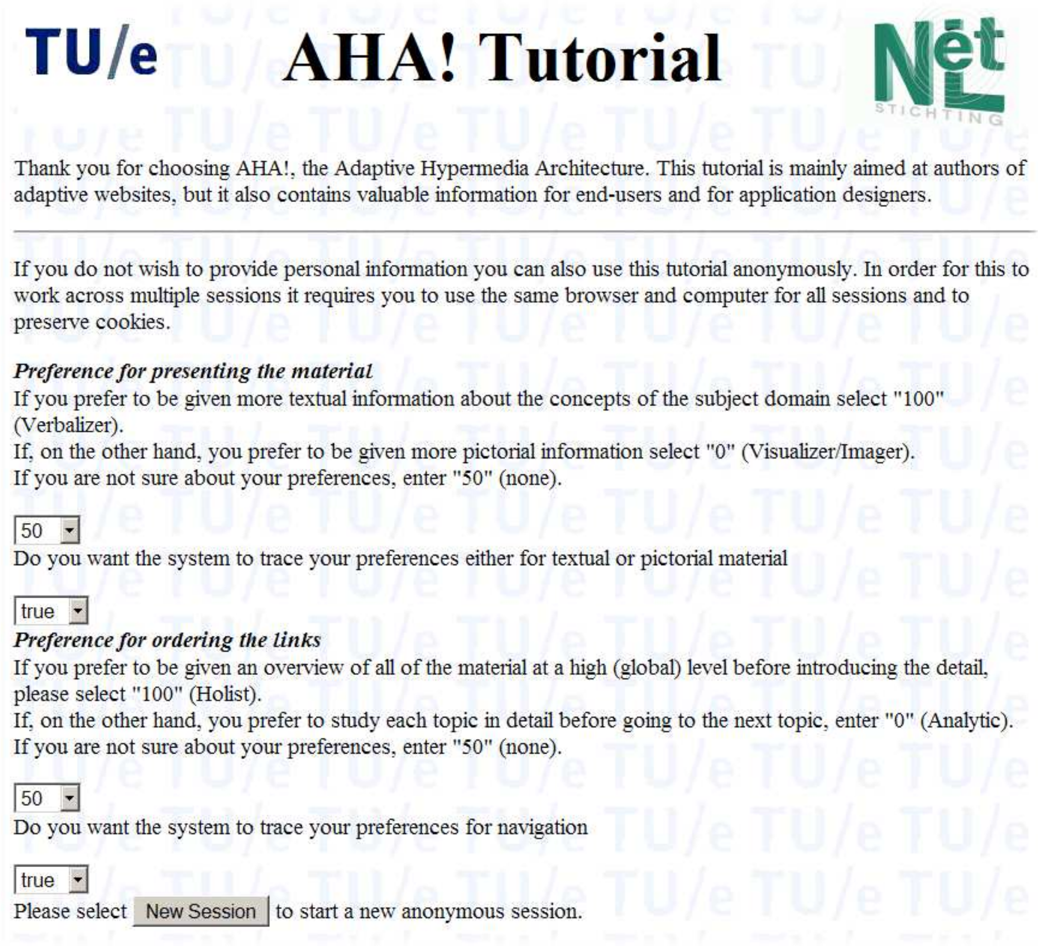
6.2.3.1 Visualization of the strategies for verbalizer versus imager and global versus analytic style

Figure 6.19 shows the presentation of the "AHAtutorialLS" material to a user with an imager and global style.

Based on the visual preference, the topic about the "adaptation performed by AHA!" is presented through an image. In the left frame the user can see the table of contents. There, links to topics are annotated (recommended topics: blue with green bullets; not recommended: black with red bullets; recommended and visited topics: purple with white bullets) so that a user is first guided to concept pages at the same level in the hierarchy as the current concept, and afterwards to lower level concepts. In the example, after reading about the "adaptation performed by AHA!", the link to the same level topic "adaptation engine" is presented as desirable.

Figure 6.20 shows the presentation of the same application to a user with a verbalizer and analytic style.

To him/her, the "adaptation performed by AHA!" topic is presented with text. The adaptive link annotation in the table of contents is also different. After reading about the



TU/e **AHA! Tutorial** NetL STICHTING

Thank you for choosing AHA!, the Adaptive Hypermedia Architecture. This tutorial is mainly aimed at authors of adaptive websites, but it also contains valuable information for end-users and for application designers.

If you do not wish to provide personal information you can also use this tutorial anonymously. In order for this to work across multiple sessions it requires you to use the same browser and computer for all sessions and to preserve cookies.

Preference for presenting the material
If you prefer to be given more textual information about the concepts of the subject domain select "100" (Verbalizer).
If, on the other hand, you prefer to be given more pictorial information select "0" (Visualizer/Imager).
If you are not sure about your preferences, enter "50" (none).

50
Do you want the system to trace your preferences either for textual or pictorial material

true
Preference for ordering the links
If you prefer to be given an overview of all of the material at a high (global) level before introducing the detail, please select "100" (Holist).
If, on the other hand, you prefer to study each topic in detail before going to the next topic, enter "0" (Analytic).
If you are not sure about your preferences, enter "50" (none).

50
Do you want the system to trace your preferences for navigation

true
Please select New Session to start a new anonymous session.

Figure 6.18: Registration form for AHAtutorialLS

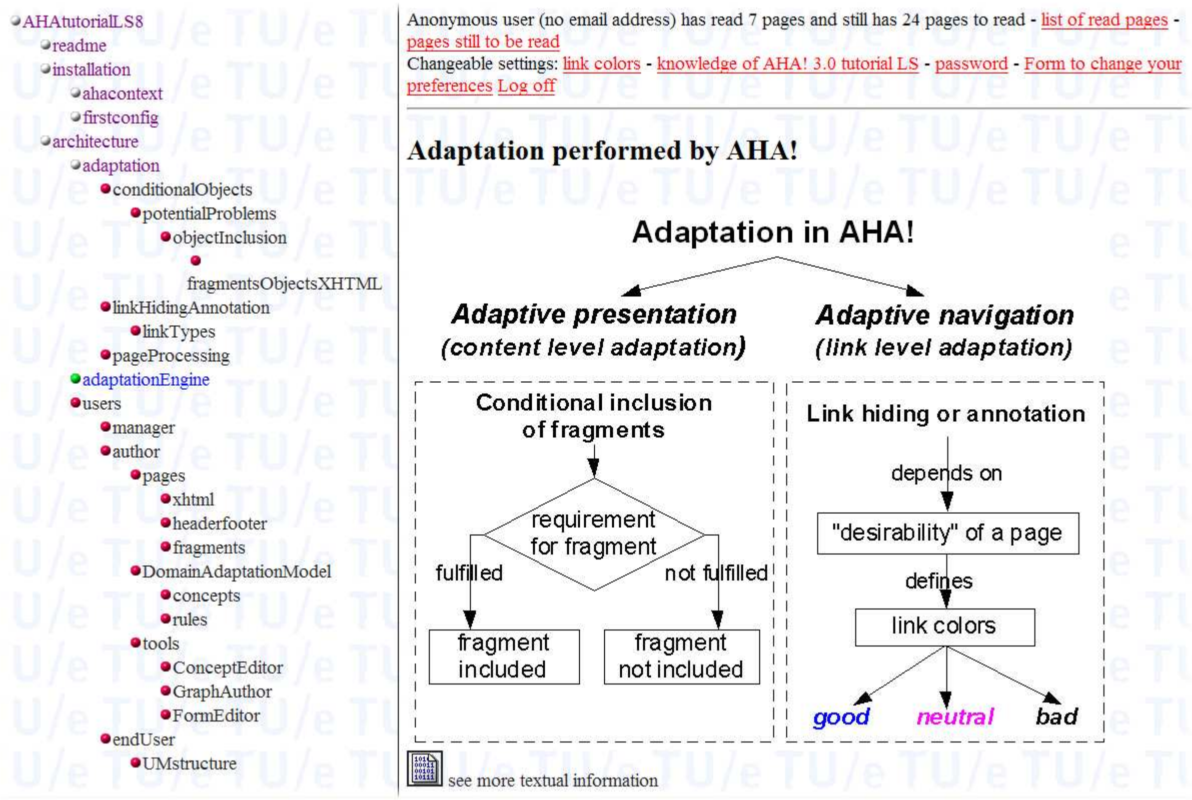


Figure 6.19: Presentation of the application to the user with imager and global styles

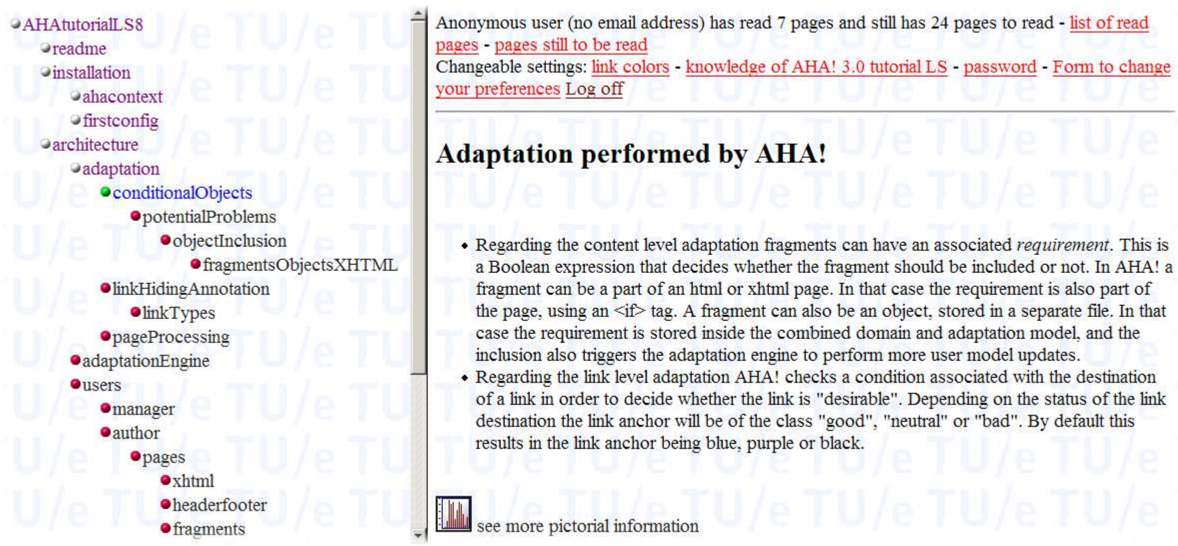


Figure 6.20: Presentation of the application to the user with verbalizer and analytic styles

current topic the user is guided towards more details on the same topic; therefore the link to the page on "conditional objects" is annotated as desirable.

6.2.3.2 Visualization of the strategy for inferring preferences for image or text

If the user does not choose any preference via the registration form (see figure 6.21) the system presents all links in the left frame as desirable. For topics that can be presented differently for users with visual or textual preference, a "default" representation is shown.

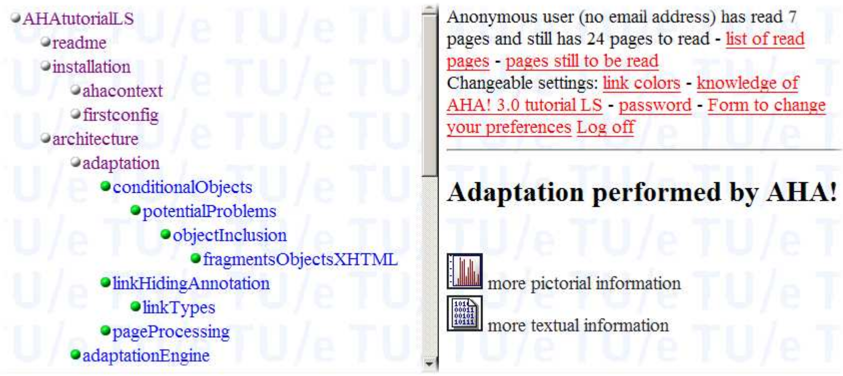


Figure 6.21: Presentation of the application to the user with no preferences specified through the registration form

If the learner lets the system trace his/her preferences in the "AHAtutorialLS", after a number of browsing steps a preference for text versus image or for navigation order can be identified and a notification about the inferred preference will be shown to the user. In figure 6.22 the system gives a notification about the identified preference for pictorial information. AHA! also allows learners to change their user model settings via special forms. Therefore, if a learner does not agree with the system's assumptions about his/her preferences, he/she can inspect his/her user model and make changes to it.

6.2.3.3 Visualization of the strategy for activist versus reflector

Figure 6.23 shows the presentation of the "WritingApplets" material to a user with the activist style. When the user starts this application (s)he is presented with some information about the "WritingApplets" concept. The representations of that concept are shown in a sorted list on the left, starting with the recommended ones. The system advises the user to follow a link to "AppletActivity" — a concept describing an activity. When the user follows that link the order of the links on the left is changed showing "AppletExample" describing an example as the next recommended concept (and as specified in the activist versus reflector strategy definition, see figure 6.11). Similarly, after accessing "AppletExample", the next recommended topic will be "AppletExplanation" and finally after accessing "AppletExample" the system will recommend "AppletTheory".

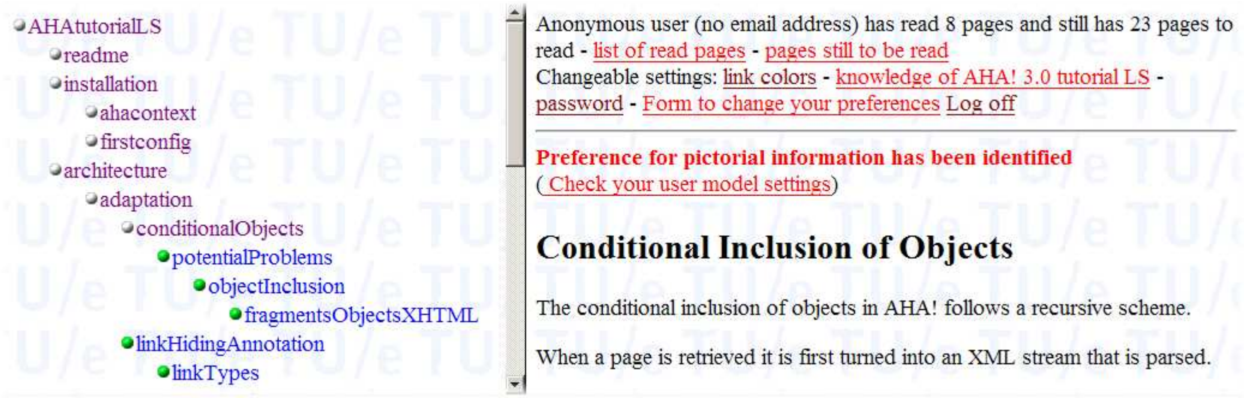


Figure 6.22: Notifying the user about the inferred preference

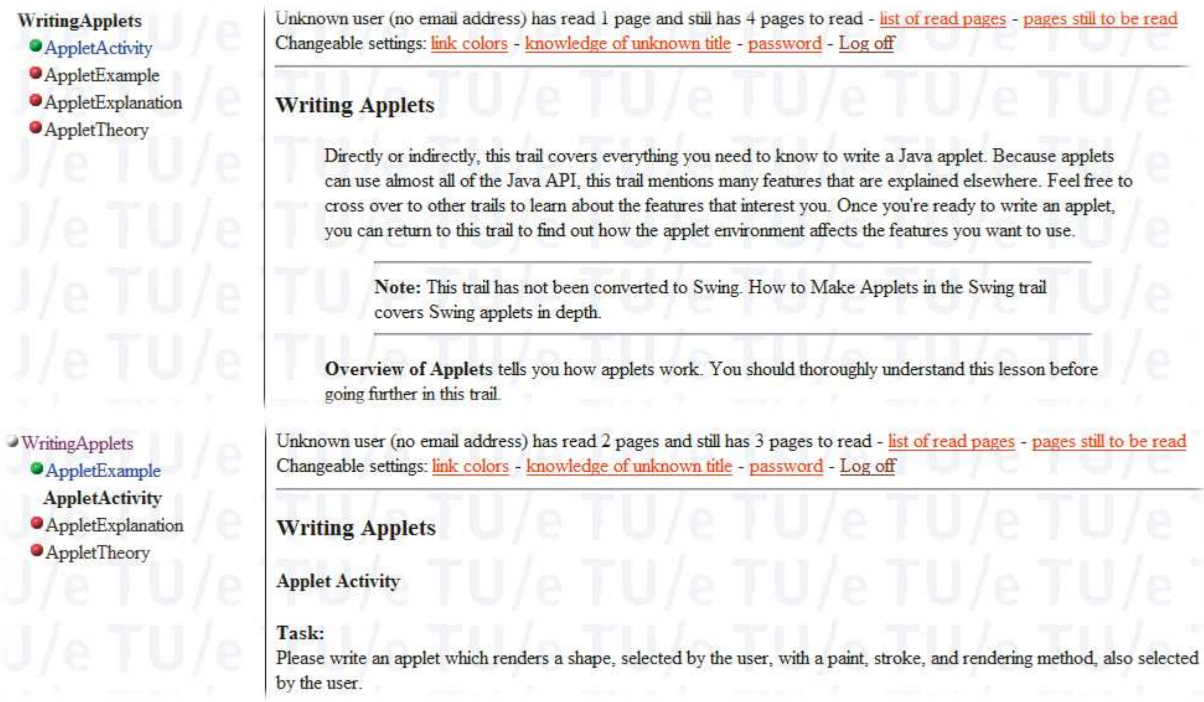


Figure 6.23: Presentation of the application to the user with activist style

Figure 6.24 shows the presentation of the "WritingApplets" material to a user with the reflector style. First, the user is first presented with the information about the "WritingApplets" concept; however the order of the links on the left is different than for an activist. The first recommended (sub-)concept is "AppletExample". After accessing that concept the next recommended one will be "AppletExplanation" and so on.

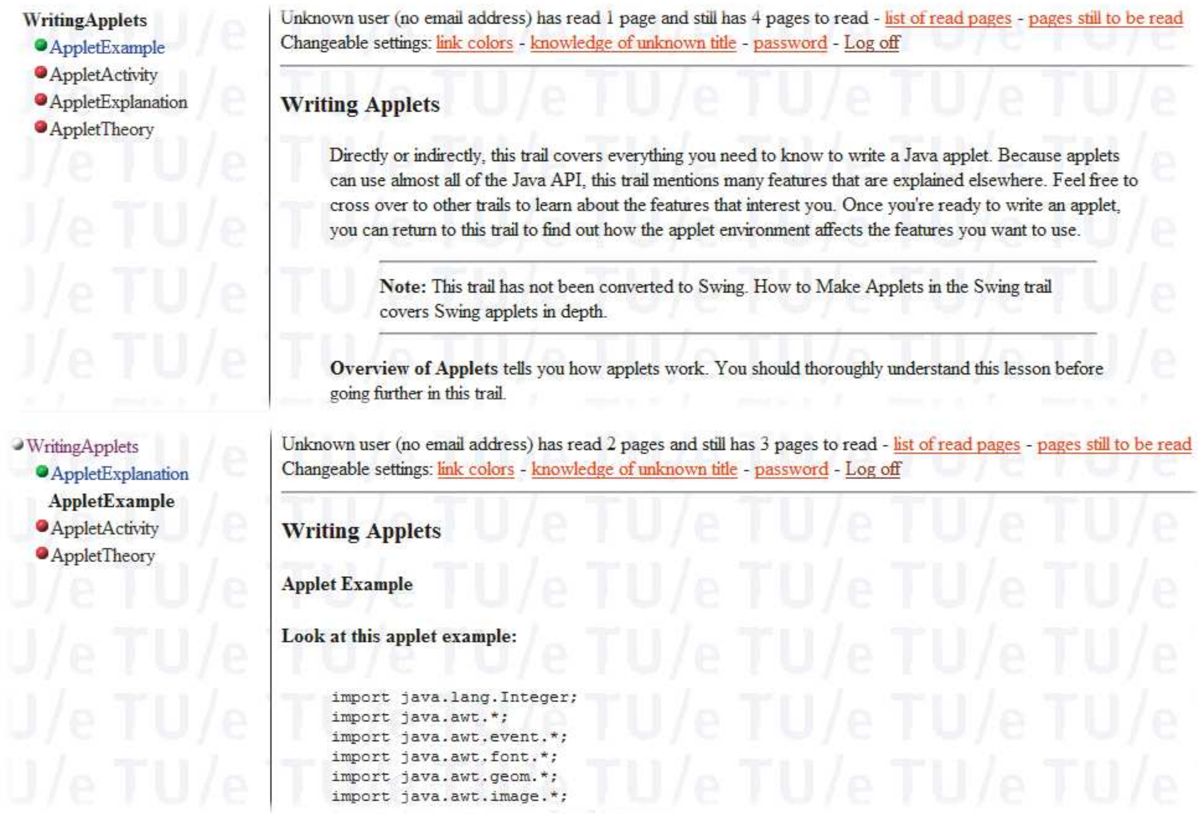


Figure 6.24: Presentation of the application to the user with reflector style

6.2.4 Summary of the sub-section

The previous sections of this chapter help us to answer research question 3 of this dissertation "Can the adaptation that is required for learning styles be realized through the AHA! system?". Let us have a look at the research question 3a first: "Can the AHA! engine perform the adaptation that is required for different learning styles?"

In section 5.3 of chapter 5 we provide the classification of the learning styles strategies as follows:

1. instructional strategies such as

    - selection of items,

- ordering information or providing different navigation paths and
- providing learners with navigational support tools;

2. instructional meta-strategies such as

- preference for certain items and
- preference for items order and navigation paths.

Through this chapter we have been addressing all these types of strategies except for providing learners with navigational support tools. AHA! provides navigational support through different menu-like views, that can be seen in Figures 6.19, 6.20, 6.21, 6.22, 6.23 and 6.24. So at least for some strategies AHA!'s standard layout mechanism with predifined navigation views already offers sufficient navigation adaptation.

Based on the information from section 6.2.1 let us revise how different types of strategies are translated to AHA!:

- selection of items — application of this type of strategy adds extra concepts and pages and attributes of concepts. The strategy also influences the showability attribute of parent concepts that can be represented by different items under various conditions.

- ordering information or providing different navigation paths — this type of strategy adds prerequisites to the domain model concepts.

- meta-strategies such as preference for certain items, items order and navigation paths add extra actions to the attributes of the domain model concepts.

Thus these types of strategies require only certain changes in the structure of the domain model concepts and adding extra pages to the application content. However these changes can be handled in a standard way by the AHA! engine. As explained in section 6.2.1 providing link sorting requires adding a new view however this is done in the layout part of AHA! which makes adding extensions easy. What is important is that the representation of the discussed strategies in AHA! applications does not require extensions to the AHA! core *engine* showing its generality.

Therefore an answer to research question 3a is that the AHA! engine can provide content and links adaptation that is required for certain learning styles, and it can also provide monitoring capabilities for inferring preferences corresponding to those styles.

However, there is one point where the support in AHA! is lacking: the layout adaptation. Some limited adaptation is possible, through page variants and through extensions of AHA! with new views, but real layout adaptation as needed to fully support e. g., the field-dependent versus field-independent styles would require extensions that go beyond the scope of this dissertation.

The second question 3b is "Can the adaptation to learning styles be created using the existing "standard" AHA! authoring tools?" In [Stash and De Bra, 2004] and [Stash et al., 2004] we show that we can create adaptation to such learning styles as verbalizer versus imager, field-dependent versus field-independent and activist versus reflector style using the AHA! standard tool Graph Author. The tool allows for the definition and application of generic adaptation rules that can be connected to various styles. However, as we already mentioned in section 6.1.1, though using these relationships can already save a lot of authoring work because the author does not need to specify the low-level adaptation rules for separate concepts, (s)he still has to draw all these relationships between separate concepts in the application graph in the Graph Author tool.

To deal with the authoring problem we defined in this chapter the learning styles adaptation language LAG-XLS for defining strategies. To be able to apply the strategies defined in this language to AHA! applications, we first had to extend the Graph Author with the possibility of strategies selection and application. Second we had to extend the Graph Author with the possibility of recognizing the variables used in the strategies such as "previousSibling", "ancestorAt{i}", "parent.firstChild" in addition to other variables such as "parent", "concept" that can be already recognized by the tool.

Thus the answer to research question 3b is: we can create adaptation to learning styles using the existing "standard" AHA! authoring tool Graph Author. However this will not be an efficient process since it will require a lot of authoring work. Our extension to the Graph Author tool allows to save quite a lot of it.

As a summary an answer to research question 3 "Can the adaptation that is required for learning styles be realized through the AHA! system?" is: only content and links adaptation that is required for most learning styles can be realized through the AHA! system. However AHA! fails to provide layout adaptation that is required, e. g., by field-dependent versus field-independent style.

To provide adaptation to learning styles in AHA! we defined the LAG-XLS language, added an extra concept template, created a tool for authoring the strategies and extended the Graph Author with a number of new features. Thus all changes have been done "on top" of the AHA! engine. No changes to the AHA! engine itself have been done.

6.3 Incorporating LS in AHA!: Approach Validation and Evaluation

6.3.1 Approach validation: representing features of AHS providing adaptation to learning styles in AHA!

In this sub-section we look back at the AHS providing adaptation to learning styles that were presented in the previous chapter and show if and how their adaptation to learning styles can be realized in AHA! using the standard AHA! features and our learning styles extension to AHA!. We show which elements of LAG-XLS can be used to specify corresponding types of adaptation that are present in certain systems. The results of this validation are presented in table 6.2.

AHS providing adaptation to LS	**Implementation of corresponding adaptive behavior in AHA!**
ACE	Sequencing of learning material using "sort" element of LAG-XLS
AES-CS	- Adaptive presentation can be realized in the same way as in AES-CS through conditional text and page variants representations, when using LAG-XLS page variants can be prepared and selected based on "select" element; - Link annotation of recommended and not ready to be learned concepts; - Annotation using colored checkmarks applied in AES-CS to show several levels of students' knowledge is currently not possible in AHA!; - Direct guidance is currently a bit tricky in AHA!, requires a lot of authoring effort; - Maximum versus minimum feedback to assessment questions — can be provided through AHA! testing component
APeLS	Selection and sequencing of learning material using "select" and "sort" elements of LAG-XLS
CAMELEON	- Adaptive presentation using "select" element of LAG-XLS; - Adaptive navigation support through link removal
CS388	Sorting of learning material using "sort" element of LAG-XLS
INSPIRE	Selection of learning material using "select" element of LAG-XLS
iWeaver	Selection of learning material using "select" element of LAG-XLS
MANIC	- The stretchtext technique applied in MANIC for adaptive presentation is currently not realized in AHA! (but can be obtained through AJAX), adaptation to learners with preferences for graphic versus textual information can be realized instead through conditional objects in AHA!, using "select" element of LAG-XLS; - Determining learner's preferences for information types, can be realized as a variation of the predefined strategy for text versus image preference
Tangow	Sequencing of learning material using "sort" element of LAG-XLS

Table 6.2: Representing features of AHS providing adaptation to LS in AHA!

As can be seen from table 6.2 we can use the “select” and “sort” element of LAG-XLS in the corresponding strategies for representation of the types of adaptation used in most of the systems. We can also provide the monitoring strategy for inferring preferences in terms of explanations, examples and graphics as MANIC does.

There are several examples in the table that require clarification.

To implement AES-CS adaptive presentation through LAG-XLS in AHA! it is possible to define different page variants with a different layout and containing different navigational tools by specifying these variants as children “object” concepts of a parent “page” concept. When the content of a parent page is empty the contents of its child will be inserted using the “select” element of LAG-XLS. Thus learners with different styles can be presented with a corresponding layout and navigational tools through different versions of the pages. In this way AHA! follows the approach adopted in AES-CS and also provides a solution for “imitating” the layout adaptation in AHA! when in fact layout adaptation is currently not possible in AHA!.

Using a different color scheme for links annotation than in most of AHA! application — “gray” for not ready to be learned concepts and “blue” color as in most AHA! applications for recommended concepts — AHA! can replicate link annotation in AES-CS.

Currently AHA! has three types of links — good, neutral and bad. Neutral link refers to a concept that has been visited when recommended thus completely understood. However AHA! does not recognize the state of learner’s knowledge on a concept — “know”, “learned” and “well-learned” like AES-CS does.

Another feature of AES-CS — direct guidance — can be achieved in AHA! through conditional content but in quite a laborous way that requires a lot of authoring effort (see section 2.4 of chapter 2).

Maximum versus minimum feedback to assessment questions can be provided through the AHA! testing component. According to [Kulhavy and Stock, 1989], effective feedback provides the learner with two types of information: verification (a judgement of whether the learner’s answer is correct/incorrect) and elaboration (relevant cues to guide the learner towards a correct answer). Tests in AHA! can be created using the Test Editor tool [Romero et al., 2006]. This tool facilitates the development and maintenance of different types of XML-based multiple-choice tests for using in Web-based education systems. It supports classic and adaptive tests.

In AHA! tests results *verification* is provided through setting a flag for each answer to a question, indicating whether this answer is correct. *Elaboration* on the tests results can be provided by adding explanations to answers.

To provide maximum versus minimum feedback to assessment questions AHA! author can prepare two different versions of a test — one with extensive explanations for incorrect answers giving relevant cues to guide the learner towards a correct answer and another one containing minimum explanations. Then the author has to create a test concept associated with these two versions of a test. Based on the user’s learning style a version of the test with maximum or minimum feedback will be presented to the user.

MANIC uses the stretchtext technique to provide adaptation to learners with preferences for graphic versus textual information. As explained in section 3.1.3 of chapter 3 this technique cannot be directly realized in AHA! but can be obtained by using AJAX technology. Adaptation to learners with preferences for graphic versus textual information can be realized more easily in AHA! by using conditional objects, which is done by using the “select” element of LAG-XLS.

To summarize, we have seen that standard AHA! and our learning styles extension provide ways for representing almost all features that exist in other systems providing adaptation to learning styles. The tricky ones that are not (fully) implemented in the standard AHA! are the direct guidance and stretchtext techniques.

6.3.2 Approach evaluation: experiment with students

6.3.2.1 Evaluation Settings

To evaluate our approach we tested the application of (meta-)instructional adaptation strategies created in LAG-XLS and applied to AHA! within an Adaptive Hypermedia (AH) course (available at http://wwwis.win.tue.nl/2ID20/). The course was given to a group of 34 students composed of 4th year undergraduate students studying Computer Science, combined with 1st year Masters students in Business Information Systems at the Eindhoven University of Technology. This experiment targeted mostly issues such as understanding of how the system works, authoring ease and satisfaction with the resulting presentation.

6.3.2.2 The Experimental Assignment

The experimental steps of the LAG-XLS assignment were as follows (see the full contents of the assignment in Appendix A).

1. The students had to perform the assignment in groups of 2-3 people in a period of 4 weeks.

2. They had to install the AHA! system version that supports learning styles on their notebooks. The distribution contained two example applications — “AHAtutorialLS” and “WritingApplets” - and a number of strategies to apply:

   - Two instructional strategies — verbalizer versus imager (VerbalizerVersusImager) and global versus analytic style (GlobalVersusAnalytic) — as well as two monitoring strategies — text versus image preference (TextVersusImagePreference) and breadth-first versus depth-first preference (BFvsDFpreference) — had to be applied to the “AHAtutorialLS” application. The instructional strategy activist versus reflector (ActivistVersusReflector) had to be applied to the “WritingApplets” example.
   - The students were asked to work with the system as authors as well as end users. As authors they used the Graph Author tool — to see the concept structure of the courses and to select strategies to apply to a particular course. As end

users they had to experience the result of applying the strategies, while browsing through the course. They had to analyze how the same course is presented with different preference settings corresponding to different learning styles, as well as with the option of automatic preference tracing.

3. After the above steps were completed, the students had to fill out a questionnaire (see Appendix A) to report on their experience of working with the system.

4. The students were also asked to fill out the Felder-Soloman "Index of Learning Styles Questionnaire" (ILS) [Felder and Soloman]. This psychological questionnaire maps a set of 44 questions over 4 dimensions representing learning preferences and styles. For our experiment dimensions of interest are represented by the values extracted for such learning styles as active versus reflective, visual versus verbal, sequential versus global. The aim was to examine if the students' preferred settings for working with the applications (as selected by them when using AHA!) corresponded to the learning styles revealed by the ILS questionnaire. Moreover, we aimed at checking whether the system's inferred preferences matched those of the ILS questionnaire.

5. Finally, after experimenting and analyzing the existing strategies, the students were asked to create their own strategies or a variation of the existing (predefined) strategies in the LAG-XLS language and to apply them in the provided applications.

6.3.2.3 Experimental quantitative results

Figure 6.25 presents the comparison of students' stated preferences corresponding to learning styles (based on the provided learning styles description and learning styles representation in the given AHA! applications, shown as the first columns for each style) and the learning styles as identified by the ILS questionnaire (second column for each style).

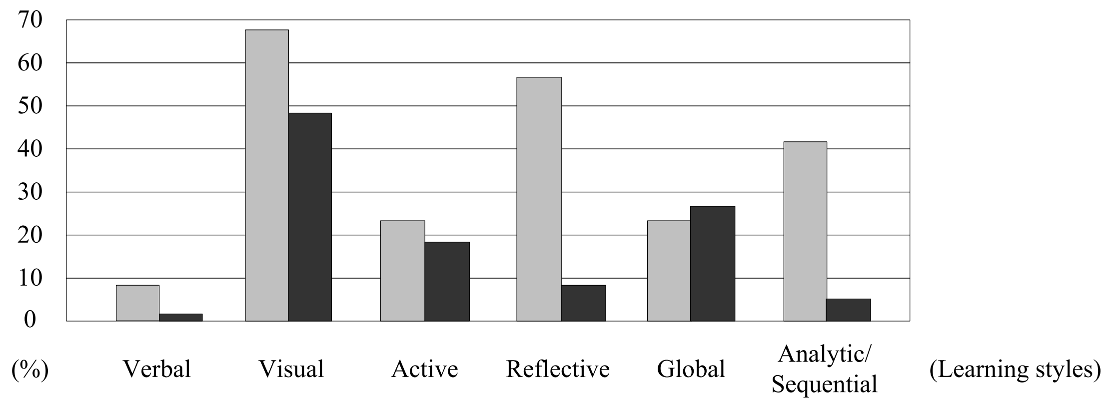


Figure 6.25: Comparison: own stated preferences and ILS questionnaire results (expressed in percentage)

The results show quite a significant difference between the stated preferences and the learning styles identified by ILS. This difference is especially noticeable in active (24%/19%) versus reflective (56%/9%) and global (24%/26%) versus analytic (41%/5%, equivalent in this assignment with "sequential") group. In the ILS, the active and global tendency is stronger, whereas in actual use, the reflective and analytic tendency dominates. The students' comments (presented later) partially explain this gap between theory and praxis. One point in which both stated preferences and ILS questionnaire results coincide is the students' strong image over text preference. However, their intensity is, again, different in praxis and theory (9%/1% for verbal style and 68%/49% for visual style).

The students' prior knowledge is shown in figure 6.26. As most of them are computer science students, unsurprisingly their XML knowledge (79% claiming that knowledge) was far greater than their prior knowledge on learning styles (24%). The fact that most students had never heard of learning styles before may be another explanation for the fluctuating results on learning preferences.

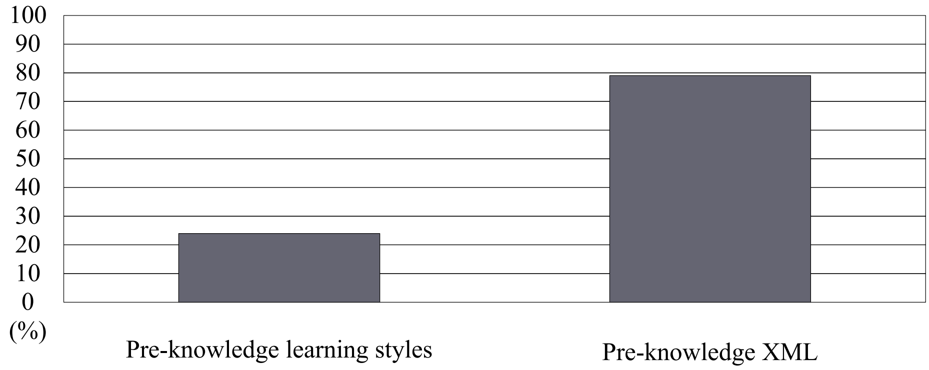


Figure 6.26: Students' pre-knowledge of learning styles and XML (in percentage)

Figure 6.27 depicts the students' general impression of their first encounter of learning styles in combination with adaptive hypermedia. Students considered the implementation of adaptive instructional strategies and (monitoring) meta-strategies useful for adaptive educational systems (82%). Less strong, but still positive was their conviction about this experimental process being of a pleasant nature (67%). Most of the students having reservations also gave some justifications, as is shown and discussed later. Figure 6.27 also shows that a majority of the students considered the work easy, although the percentage of students with that opinion is slightly lower (54%). This difference shows that although students realized the necessity and importance of adaptive strategies in adaptive hypermedia and enjoyed the challenging programming work they did not consider it trivial. Therefore reuse of ready-made, custom-designed strategies is necessary to be made available to adaptive hypermedia authors to reduce creation time and costs.

Figure 6.28 shows the average declared percentage of understanding and problems that students encountered. An ideal distribution should create a filled pentagon. A good distribution should at least have all the corners of the pentagon at values above 0.5, as

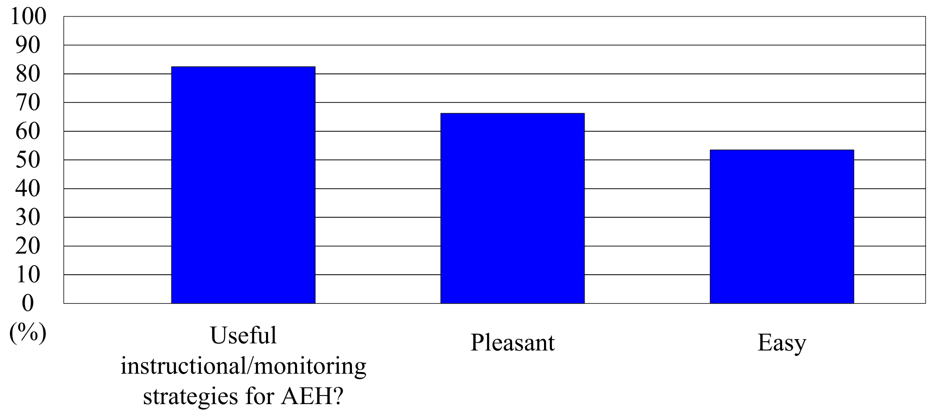


Figure 6.27: Overall impression of instructional strategies and experiments (in percentage)

is almost the case here. The students understood the application strategies (77%) — important as the core of the LAG-XLS language understanding — and were greatly satisfied with the presentations (76%). They understood the AHA! Graph author very well (88%). However creation of their own strategies was the most difficult problem (only 47% had no problems with editing). When they figured out editing, their strategy changes worked well (75%).

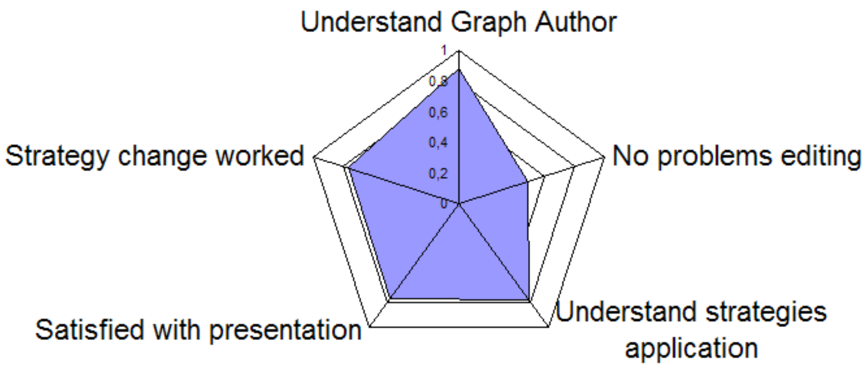


Figure 6.28: Understanding the system and working with it (in percentage)

Figure 6.29 presents the results of the comparison whether students' preferred settings and system induced preferences were the same as the ILS questionnaire results.

The preferred settings were the same as the ILS questionnaire results in 87% for verbalizer versus imager strategy, 73% for global versus analytic and 67% for activist versus reflector strategies. The induced preferences were the same as the ILS questionnaire results in 85% for breadth-first versus depth-first preference and 79% for text versus image preference meta-strategies.

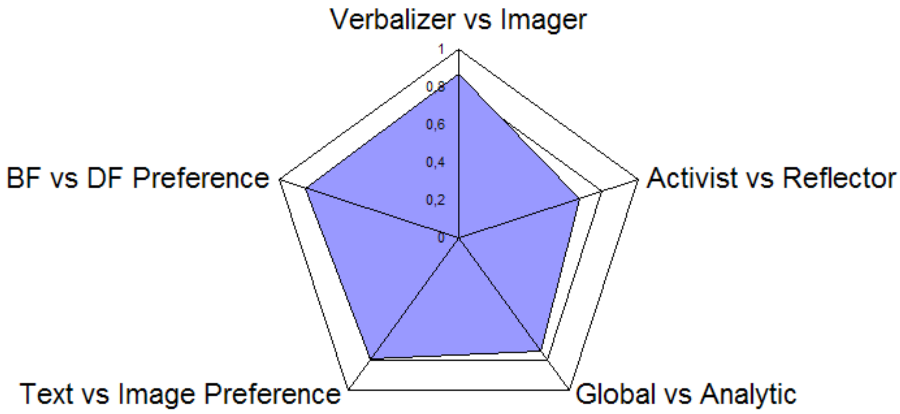


Figure 6.29: Students comparison of questionnaire results versus praxis results whilst working with the system (praxis via questionnaires versus praxis deduced by system - in percentage)

6.3.2.4 Experimental Qualitative Results

The students were asked to also detail their own judgements and explain their understanding of the process of using the AHA! system together with the implementation of learning styles via the LAG-XLS language. Below we provide some sample comments. We kept the original students' spelling.

1. When asked if they thought that application of different instructional/monitoring strategies for educational adaptive hypermedia is useful, students replied:

   "Yes", because:

   - "I believe that the correct application of learning styles can be a good aid in studying. Presenting information in a user preferred form makes the user work and study more efficiently."
   - "In this way, different types of persons can get the same information presented in a way that is most helpful to them for learning."
   - "Since each person is unique, and does his/her best when anything is tailored to his/her unique needs. Perception skills vary from person to person, so if it is possible to give each person, the best possible method of learning suited to him/her, it is the best possible educational method."
   - "Adapting a big amount of information to the best way the user perceives could lead to saved time and a better understanding of the studied problem."
   - "It can be helpful to show the information for a student in the, for that student, best way. This may help to learn the material faster."

- "When you know what kind of person you are (or the computer knows) it saves time, because you don't have to look for what you want, you automatically get it. And if you want more or other information on some subject then you normally would, you can just click on a link or something and still get it."
- "In order to fit educational adaptive hypermedia to the purposes of each respective user, such strategies can be used to adapt as much as possible to the needs of such a user. In some cases a user is not at all aware of its capabilities and which learning styles he/she mostly prefers. With such strategies a user is guided through the learning material as best as possible."
- "Every student has his own way of learning. In a classroom there are several methods that can be used to learn the material. Part of the students prefer the classes themselves, others refer to written course material. And of course there is the option to discuss uncertainties. In an adaptive hypermedia environment the situation is different. The data is semi-static and no real discussion is possible. In order to satisfy all the students taking the course there has to be a way to customize the environment that suits the learning style of the students, because this material is all that is available to them."

One of the students having said "Yes" had certain reservations:

- "There are a few things to say in this matter. The question remains if users always have a similar learning style. I can come up with a lot of different subjects that I have studied, while using a total different technique to master them. I think however, that instructional/monitoring strategies for educational adaptive hypermedia are indeed useful for small tutorials, where information can be presented using a guide. It is probably also very useful for people who are not used to study a specific topic."

  Here, the student correctly noticed that learning styles may vary in different subject domains (as has been discussed in section 4.4 of chapter 4).

"No", because:

- "Even when you only click on a link, the page is registered as known. In this way you can fool the system easily. Only when links that are not yet available are really hidden and there is a timer before a page is registered as known, this strategies can be useful. At this moment, I don't think that my learning speed and performance is changed by using different strategies."
- "It is very easy to fool the system. The system doesn't check if the content is understood by the reader."
- "In theory the adaptive hypermedia could adjust to the preferred learning style of the student and ensure the most benefit from the learning experience. However, in practice, I believe the system has too many weak chains to be successful; it depends highly on competent authors, a wide availability of learning material

in many different forms, and the ability of a computer program to (correctly) reason about a human behind the terminal."

Here the student correctly identifies the authoring problem: adaptive hypermedia is more time-consuming and costly than regular hypermedia — the price paid for adaptation [Brusilovsky, 2003].

To summarize, the students considered the idea of learning styles application in the adaptive hypermedia a useful endeavor because of issues such as efficiency, better understanding of a subject problem and saved time. Those students who were sceptical about this idea explained their answers by the fact that the system does not check whether the user really understood the content and by recognizing the authoring problem.

2. In explaining why the ILS questionnaire results were different from the stated preferences the students answered:

   - "My pictorial preference in the Tutorial was not representative of my general preferences (which were shown by the questionnaire). In this specific Tutorial application the pictures however were so good that these were preferred by me."
   - "I generally like to see the global picture first and then go into the details. However in the tutorial, this raises a problem for me. If I read the high level concepts first and then go into the details, I have forgotten what the high level contents were when reaching its details and then I have to read back into it. That is annoying, so I prefer to read depth-first. So actually I'm quite unsure what I prefer. Maybe I do prefer depth first. It's a bit hard to tell really."

     The described problem may be caused by the fact that the example application "AHAtutorialLS" was created by the authors of the system who might not have enough psychological knowledge about how to correctly structure the application in order to support the global and analytic learning styles. However the system provides the necessary functionality to present the application either in breadth-first or in depth-first order as recommended by the psychological research to support global and analytic LS correspondingly.
   - "I like "big picture" overviews, but in practice I usually read a text front-to-back and then attempt to digest the contents. So that will probably explain the difference."
   - "The settings made in the AHAtutorialLS course were chosen for this very specific case while the results from the questions at the website on learning styles are somewhat more general. They apply to more aspects and disciplines than just one (electronic) course. As you can see, the preferences are just slightly off."
   - "The answers in the questionnaire don't exactly apply on the tutorial and applet learning in AHA. For example, when learning I would rather see an example,

read theory and then try myself, but when doing some sport I'd rather try it myself."

- "The test shows that I prefer Activist a little bit more than Reflector. It depends on the situation and the examples that are shown what is more useful to me. In my case a good example works better than an exercise, but otherwise working on an exercise is a good start to learn about a topic."
- "Maybe I try to learn in a Reflector way, but that is not the best way for me to attain information, as can be seen from the questionnaire output."
- "The test results only show a slight preference for Activist and Global. I don't think I have a very strong preference for either one and I found it difficult to choose on a lot of questions because both answers seemed appropriate. The results in Table 1 are also due to small things so I think the difference is due to me not really having a strong preference either way so sometimes a test will show one and sometimes the other."

As the student correctly identified himself the problem lies in the fact that we considered only bipolar dimensions of learning styles in our example applications but we did not take into account various gradations of learning styles preferences as can be recognized by the ILS questionnaire.

To summarize, the differences between the ILS questionnaire results and the stated preferences can be explained by the fact that the provided example applications were not the "real use" ones (concerning the content: correctness of structure and various concepts' representations) but these were the examples mostly targeted at showing possible types of adaptation to learning styles. This caused a confusion among a number of students.

3. In explaining why the induced preferences were different from the results of the ILS questionnaire several students referred to the answers for the difference between the ILS questionnaire results and stated preferences. In addition some students answered:

   - "... since the system has to induce a definite preference for BF vs DF and my personal preference is inbetween them, the system can never derive my lack of preference."
   - "In AHA I have got only a binary choice, the result on the Internet is not binary. So I cannot compare them. As a logical result, the answer is No in this case."
   - "I still prefer the BF way but the system traced a depth view for me. I think this has more to do with the structure of the tutorial. (I don't read only the first pages of chapters but I certainly don't go straight away to depth either). The tutorial goes at first a bit in depth and then you can go in breadth, so if you prefer to go a bit deeper at first, it already traces a depth view."
   - "I think this is related to the subject to be studied. Subjects I have no knowledge of, I'd rather read some global information before diving into details. Subjects

I do know something about I like to plunge the deep and go into detail. So it's case related."

Again, the problem lies in structuring the applications and in the fact that gradations of learning styles were not considered in the applied strategies.

4. When asked about other strategies they would like to apply but were not able to express in LAG-XLS, students replied:

    - "No, I can't think of any other strategies. The strategies provided are sufficient for me. The tutorial reacted to my preferences as I expected."
    - "developing an entirely new strategy is impossible without completely altering the entire Tutorial application ... We looked at the XML-files and of course we could make small alterations which change the number of steps after which a preference is derived, but we did not think that this was what you were looking for, since the general appearance of the system would be exactly the same, and the results are easily predicted."
    - "GoodReadingVersusFastReading — a strategy that is able to track if a reader really does an effort to study/read the educational material presented. That way the system could "warn" the user when he or she just seems to be clicking trough the material instead of actual 'learning' a matter".

      This is an interesting strategy, for its realization additional AHA! programming is necessary to check the time spent on reading AHA! application pages.

    Several students proposed strategies that include preliminary or intermediate tests in the course. These kinds of strategies can be realized using the standard AHA! testing component without the need for LAG-XLS. Other students were only able to create variations of the existing strategies by using different names for presentation items and by increasing/decreasing the number of steps required by the monitoring strategies to achieve a threshold. The students did not come up with any completely new strategies.

5. When asked to give some more suggestions for possible improvements of the current LAG-XLS implementation, students answered:

    - "The system should have a time table to make sure the pages are read and not only clicked. The will improve the quality of the decision making which strategy should be applied."
    - "The interface of graph authoring tool could be more user-friendly."
    - Most students asked to make the AHA! installation easier; to have more help explaining the effect and application of strategies.

6.3.2.5 Summary of approach evaluation with the students

The evaluation results (presented in [Stash et al., 2006a,b]) show that students understood the process of learning styles application in AHA! and liked being involved in it. They were able to work in the authoring environment with the predefined example applications and strategies as authors and as end-users they were quite satisfied with the resulting presentations. It is very reassuring that our students understood the basics of learning styles application, although they are computer science students, with little or no knowledge in this field prior to the course.

The most difficult part of the assignment was the evaluation of the LAG-XLS expressivity. The students could mainly think of some modifications to the existing strategies. However they were not able to create new strategies from scratch. A possible explanation for that could be the short time they had; another one, the fact that the problems with the system installation detracted from the quick application of the potential of the language; finally, it might just be possible that they were aiming too high (see comment on what the teacher might want).

It is clear that the creation process of adaptive behavior in itself requires a lot of psychological and/or pedagogical knowledge. As we are not psychologists, the main aim of our research is to allow the authors with experience in pedagogical psychology to design different types of strategies and apply these strategies to the applications. The question about how to structure the application and organization of the materials to correctly suit different learning styles is left for the author of the application or psychologist.

We therefore involved a number of learning styles experts to validate our tool. The results of this validation are presented in the following section.

6.3.3 Approach validation with learning styles experts

6.3.3.1 The Validation Settings

We asked four learning styles experts to give their opinion about our system. Three of them were asked to experiment with the system online and a face to face meeting was arranged with one learning styles expert.

For the online experiment, the experts were provided with the document containing the following information: brief description of AHA!, types of strategies that can be created and applied in AHA!, description of the "AHAtutorialLS" and "WritingApplets" examples (the same applications as the ones used in the experiment with the students), description of the strategies application in the Graph Author tool. The document also provided the online links to the example applications and the Graph Author. Similar to the experiment with the students, the LS experts were asked to work with the system both as authors and end-users. However, the focus of this experiment was on the issues such as satisfaction with the effect of the predefined strategies and the expressivity of our language for strategy definition — the ability to make modifications to the predefined strategies or create completely new strategies.

After experimenting with the system, the LS experts were asked to fill out the questionnaire shown in appendix B.

During a face to face meeting with one learning expert a local version of the system was shown. The same questions as in the questionnaire in appendix B were asked.

We summarize the feedback provided by the learning styles experts below.

6.3.3.2 Experimental Qualitative Results

- *Previous experience with adaptive hypermedia.* Only one learning styles expert worked in the field of adaptive hypermedia before. The others had little or no previous experience.

- *Considerations about the idea of learning styles application in adaptive hypermedia.* The learning styles experts found this an appealing idea:

  - "I like the principle of adaptive instruction that you are seeking to implement, and the idea of adjusting the order in which material is presented to match learning style preferences is a good one."
  - "LS are important when learning since they show what the users usually don't know about themselves. Learning is more honest."
  - "I think it is a promising idea to be able to adapt an environment to an individual's style."

  Also some were "intrigued by the idea of using someone's navigational choices to draw inferences about their learning style."

- *Understanding of the strategies application in AHA!.* The learning styles experts provided the following answers:

  - "Pretty clear and straight."
  - "I don't think I had difficulties, but I would need to spend a lot longer on this, to appreciate it completely."

  One learning styles expert was not able to access the Graph Author tool as it requires the Sun Java plugin[3] which was not installed on his computer. Therefore he was able to get information about the tool only in the description document.

- *Understanding of LAG-XLS.* The description of the types of strategies that can be expressed in LAG-XLS seemed to be quite clear as well.

- *Satisfaction with the effect of the predefined strategies.* The learning styles experts had different opinions about the effect of the predefined strategies. One expert expressed high level of satisfaction with the imager versus verbalizer and global versus analytic

[3]http://java.sun.com/products/plugin/

strategies and medium level of satisfaction with activist versus reflector strategy. On the other hand, another learning styles expert was more satisfied with the idea of adjusting the order of the presented material, such as activity then reflection for active learners and the opposite for reflective ones, however suggested to be very careful about the idea of "making either-or-choices, for example, teaching something visually to visual learners and verbally to verbal learners." The problems mentioned in relation to this approach are: "learning is always more effective when information comes in both modalities, regardless of the learner's preference" and "the idea is to provide balanced instruction, not just instruction that matches preference."

These issues are addressed in AHA! by the possibility of including an alternative representation in addition to the recommended one through a link. The author can choose to provide mixed representations as well.

The differences in the opinions of the learning styles experts prove the necessity for providing the course designers with an option of creating and applying their own strategies (in addition to the predefined ones) to their courses.

- *Necessity for strategies modifications.* No suggestions for modifications have been provided except for the considerations expressed above. One learning styles expert dealing with other dimensions of learning styles than the ones we used in the examples confirmed that our predefined strategies can be reused for expressing adaptive behavior corresponding to the styles he deals with.

  Learning styles experts also suggested "including interactive self-assessments after each section" and "providing corrective feedback for incorrect answers".

  In our example applications we did not provide tests however they can be easily added in AHA! and tests' results can be used for making necessary adjustments.

- *Possibility to express necessary modifications in LAG-XLS.* No comments have been added.

- *What is possibly lacking in our approach.* One of the learning styles experts mentioned that the system uses "too many dimensions to be handy". The proposal is not to have less dimensions but providing "a multi-layer approach, showing just a few at once, depending on the level of expertise of the creator and his/her preferences." Another comment was "Could you think of designing this to feature information from other subject domains?"

- *The main strengths of the system.* One of the learning styles experts mentioned flexibility as the main strength of the system also pointing that other known tools are too restricted and single-purpose.

- *The main weaknesses of the system.* As the main weakness of our approach the learning styles experts mentioned the subject matter of the provided examples — a problem that we encountered in the experiment with the students as well. However we would not consider it as a weakness of the system since the problem is in the structure and the contents of the authored applications.

One of the learning styles experts mentioned that our tool is too complicated for the teachers who are the main community to use such tools in daily life. We cannot but agree that indeed the tool for defining learning style adaptation is difficult for teachers, and that therefore the definition of adaptation strategies should be done by pedagogy experts and that teachers should only select which learning styles they wish to provide adaptation to.

6.3.3.3 Summary of approach validation with learning styles experts

Overall the results of validation with the learning styles experts were quite positive. They liked the idea of providing adaptation to learning styles and were quite satisfied with the provided approach. The main problem they encountered was related to the structure of the sample applications — the same problem as in the experiment with the students.

As explained before, our examples were focused mostly on showing possible types of adaptation to learning styles and not on providing "real world" applications. Designing an application with support for learning styles should be carefully prepared by pedagogues and learning styles experts. As computer scientists we can only provide a tool for authoring and delivery of such applications. For this reason we did not perform any experiments to check whether learning in matched conditions with the AHA! applications increases the performance of the learners. Furthermore a lot of these kinds of experiments have been already performed by other researchers (as discussed in section 4.5 of chapter 4). In fact our tool can be used by learning styles experts to test their learning styles theories.

6.4 Summary

This chapter presented our approach to incorporating learning styles in the AHA! system version 3.0. It answers research questions 3, 3a and 3b of this dissertation.

An answer to research question 3a "Can the adaptation that is required for LS be realized through the AHA! system?" is: the AHA! engine can provide content and links adaptation that is required for certain learning styles, it can also provide monitoring capabilities for inferring preferences corresponding to those styles. However currently AHA! fails to provide layout adaptation that can be required by other learning styles, such as, e. g., field-dependent versus field-independent style. Adding the possibility for layout adaptation would require certain changes to the AHA! engine but these issues are not discussed in the current dissertation.

An answer to research question 3b "Can the adaptation to learning styles be created using the existing "standard" AHA! authoring tools?" is: we can create adaptation to learning styles using the existing "standard" AHA! authoring tool Graph Author. However this will not be an efficient process since it will require a lot of authoring work. Our extension to the Graph Author tool allows to save quite a lot of it.

As a summary an answer to research question 3 “Can the adaptation that is required for learning styles be realized through the AHA! system?” is: only content and links adaptation that is required for most learning styles can be realized through the AHA! system. However AHA! fails to provide layout adaptation that is required, e. g., by field-dependent versus field-independent style.

To provide adaptation to learning styles in AHA! we defined the LAG-XLS language, added an extra concept template, created a tool for authoring the strategies and extended the Graph Author with a number of new features. Thus all changes have been built “on top” of the AHA! engine. No modifications to the AHA! engine itself have been performed. The fact that the standard AHA! engine can deal with our learning styles extension without the need for any modifications proves the generality aspect of the created tool.

Our approach validation was quite positive in the sense of comparison AHA! with the other systems that provide adaptation to learning styles. The standard AHA! and our learning styles extension provide ways for representing almost all features that exist in other systems. Though the tricky ones that should be implemented in the standard AHA! are the direct guidance and stretchtext techniques.

The results of evaluation of the approach with the students of the Eindhoven University of Technology have shown that the students understood quite well how the system works and were satisfied with the resulting presentations. Concerning the authoring part, they understood the Graph Author tool quite well but considered creating their own strategies to be a difficult task.

The results of validation of our approach with the learning styles experts were quite positive as well. The psychologists liked the idea of learning styles adaptation in online educational courses and were quite satisfied with the provided approach. However they had reservations concerning the provided sample applications and the complexity of the adaptation strategy definition tool.

Chapter 7

Concluding Remarks

In this chapter we summarize the results of the previous chapters and give some pointers for future research.

7.1 Conclusions

In this dissertation we have studied the fields of adaptive hypermedia and learning styles and possible connections between them. We started in chapter 1 by asking five research questions about these research areas and links between them. We will now summarize our findings and give the answers that we came up with in the different chapters of this thesis.

Research Question 1: Is an AHS necessarily a special-purpose tool?
In chapter 2 we derived a set of design requirements for a general-purpose tool. In chapter 3 we showed the design and implementation of the AHA! system and checked whether and how AHA! implements the derived set of requirements.

1. *Adaptation of content, links and layout.* AHA! provides content and link adaptation, but no layout adaptation yet (section 3.1).

2. *Adaptation based on three types of information - user data, usage data and environment data.* AHA! allows for rich user model with multiple concepts and arbitrary attributes of these concepts. Through these concepts and attributes it is possible to represent various types of information and use them for adaptation purposes (section 3.2.2). However they can be used only for content and link adaptation but not for adaptation of the layout, as could be needed when adapting to different devices (with very different screen sizes).

3. *Arbitrarily many types of concept relationships.* Through the Graph Author templates the authors can create various types of concept relationships they might need for their applications (section 3.2.1).

4. *A "generic" way of defining adaptation rules.* The Graph Author tool makes this possible through the use of templates (section 3.2.1).

5. *Handling non-monotonic user model updates and cycles in adaptation rules.* Handling non-monotonic user model updates became possible in AHA! 1.0 already, but was generalized to the event-condition-action rules in AHA! 2.0, and left unchanged in version 3.0. As of AHA! 3.0 the system provides three options for handling cycles and non-confluence in adaptation rules — by making the order of rules execution visible in the Concept Editor tool, by an automatic check for cycles in the authoring tools and by limiting the number of objects included in the page (section 3.2.1).

Creating a general-purpose tool is a time consuming and difficult process. Not all of the aspects described in the list are currently provided by AHA!.

The system has the necessary functionality to provide content and link adaptation based on arbitrary user characteristics. Through the templates supported by the Graph Author the authors can specify various types of concept relationships and adaptation rules they might need in their applications. Apart from tutorials and courses AHA! has been used to create adaptive talks (adaptive Powerpoint-like presentations), adaptive information kiosks (like in one version of the IShype information for interns and master thesis students) and even a research paper [De Bra et al., 2006].

Therefore an answer to research question 1 "Is an AHS necessarily a special-purpose tool?" is "No". We have shown this by implementing the AHA! system. We have presented how an educational AHS AHA! 1.0 has been turned into a powerful tool which allows to create applications targeting various domains and not necessarily educational settings.

Research Question 2: Can we specify how an AHS should perform adaptation according to different learning styles?
An answer to this research question is "Yes". We have shown this in chapter 5 by presenting a review of recommendations from the psychological research for possible adaptive response of the system to a certain learning style and by presenting a number of AHS providing adaptation to learning styles.

Research Question 3: Can the adaptation that is required for LS be realized through the AHA! system?
This research question boils down into two sub-questions which we answer first.

Research Question 3a: Can the AHA! engine perform the adaptation that is required for different LS?
In chapter 6 we showed that the AHA! engine can provide content and links adaptation that is required for certain learning styles, it can also provide monitoring capabilities for inferring preferences corresponding to those styles. However currently AHA! fails to provide layout adaptation that can be required by other learning styles, such as, e. g., field-dependent versus field-independent style. Adding the possibility for layout adaptation would require certain changes to the AHA! engine but these issues are not discussed in the current dissertation.

Research Question 3b: Can the adaptation to LS be created using the existing ("standard") AHA! authoring tools?
Chapter 6 explained that we can create adaptation to learning styles using the existing

"standard" AHA! authoring tool Graph Author. However this will not be an efficient process since it will require a lot of authoring work. Our extension to the Graph Author tool allows to save quite a lot of it.

As a summary an answer to research question 3 is: only content and links adaptation that is required for most learning styles can be realized through the AHA! system. However AHA! fails to provide layout adaptation that is required, e. g., by field-dependent versus field-independent style.

7.2 Future work

In this dissertation we focus on two issues. One is the design and implementation of a general-purpose tool. The other one is incorporating learning styles in the created general-purpose tool.

There are several directions where extensions and improvements to the AHA! system version 3.0 presented in this thesis can be made. We outline them here:

1. An answer to research question 1 shows that not all design requirements that we derived in chapter 2 are satisfied in AHA!. Therefore in the future work we would consider targeting aspects such as:
   - providing layout adaptation,
   - exploring the adaptation techniques such as direct guidance, adaptive link sorting, adaptive link generation, map adaptation, natural language adaptation, sorting fragments, stretchtext.
2. An option for searching information in the AHA! applications can be explored. Note that there is a significant body of research into performing adaptive searching in non-adaptive information, but not into performing search (adaptive or not) in adaptive information. Searching in an adaptive application cannot be done by simply creating an index database of the content of pages. Since pages pages may conditionally include fragments and/or objects they may present different information to different users, and this affects whether they are relevant for a search or not. Also, each time the user clicks on a link from a search result page and then returns, the relevance (and ranking) of the results should be reevaluated according to the new user model state. Research into providing a search facility for AHA! may be big enough of a challenge for a whole new PhD track and thesis.
3. We would also like to improve the authoring of adaptive applications. In this relation we would consider the following issues:
   - Currently an AHA! author has to create new concept templates and concept relationship types manually which requires knowledge of the XML syntax. To help especially the non-technical author who does not have this knowledge we would consider creating an authoring tool for defining these templates.

- Authoring of the domain/adaptation model in AHA! can be done either at the high level using the Graph Author tool or at the low level using the Concept Editor. However a combination of both authoring possibilities can be necessary. We would consider creating an alternative tool that combines the advantages of both Graph Author and Concept Editor. Such a tool should allow for high level authoring as the Graph Author does. At the same time it should let authors control the lower level details such as adding or editing the specific rules as the Concept Editor does. Usage of both Graph Author and Concept Editor for these purposes is not a solution because all changes done in the Concept Editor to the application created with the Graph Author will not be visible for the Graph Author while revisiting this application.

4. As another future direction we would consider the possibility of creation and application of various types of strategies and not only the ones related to learning styles. More research is needed in order to decide whether the extensions to the authoring tools, created for learning style adaptation, are usable for other types of strategies or whether new extensions and tools should be designed.

Appendix A

Assignment for prototype evaluation

Date: 12-01-2005

Name of the course to which the assignment belongs: Adaptive Hypermedia

PhD student: Natalia Stash (e-mail: n.v.stash@tue.nl)

Please return the completed file (questions and tables) to a.i.cristea@tue.nl by 9-02-2005!

Task description:

1. Download and install AHA! version supporting learning styles from http://wwwis.win.tue.nl/ nstash/PhDThesis/experiment/ahaLS.zip.
2. Go to AHA! configuration tool http://localhost:8080/aha.
3. Select author "nvs" and look at the list of available courses and strategies. The XML files describing the strategies are located in aha/author/authorfiles/nvs directory. (You can change the author "nvs" with your own name or continue working with the system as an author with login "nvs" and password "nvs".)
4. Go to the Graph Author tool

   http://localhost:8080/aha/author/GraphAuthor/GraphAuthor.html.
5. **Instructional Strategies**: Open the course "AHAtutorialLS" and try to apply different strategies using "Strategies/Apply" option. Apply "VerbalizerVersusImager" and "GlobalVersusAnalytic" strategies.
6. Select menu item "Save/Save to AHA!" to save the concept list into the internal format. To visualize the effect of the strategy check for generated concepts or added attributes of the concepts as explained in course 5.
7. Go to the application page (http://localhost:8080/aha/AHAtutorialLS) and try how it works with different settings for the preferences (by selecting different options in the overview page).

8. Compare the different settings of preferences and write down which one you preferred most (for each strategy). Fill in the table below:

Strategy name	**List of preferred settings as selected by you**
VerbalizerVersusImager	
ActivistVerusReflector	
GlobalVersusAnalytic	
Your own strategy? (replace w. name)	

Table A.1: Your own preferences

9. Go back to the Graph Author. Open a small example "WritingApplets" and apply ActivistVersusReflector strategy. Go to http://localhost:8080/aha/WritingApplets and see what are the differences in the presentations for "Activists" and "Reflectors".

10. **Instructional Meta-Strategies**: Try the application of the monitoring strategies: Go to the Graph Author. Open the course "AHAtutorialLS". Apply the strategies "VerbalizerVersusImager" followed by "TextVersusImagePreference" and "GlobalVersusAnalytic" followed by "BFVersusDFpreference" (breadth-first versus depth-first preference). Repeat steps 5-7 by setting at step 7 options for tracing preferences to "true".

11. Count the number of steps till the system traces your preference. Write down what the system tracing result was, in the table below. For TextVersusImagePreference the number of steps is the number of accesses to textual or pictorial information. For BFVersusDFPrefences this is the number of read pages. This number can be taken from the information in the header of the application page.

Strategy name	**System induced preference**	**Is system induced preference correct? (Yes/No)**	**Number of steps till tracing occurs**
TextVersusImagePreference			
BFVersusDFPreference			
Your own strategy? (replace w. name)			

Table A.2: System traced preferences

12. **Creation of Instructional (Meta-)Strategies**: Make some changes in the strategies (XML files as at point 3) and apply them, by repeating steps 5-8. You may

select the strategy which you did not like and specify it differently using the elements from the "author/authorfiles/strategy.dtd" and variables explained in the course 5. You may also edit the existing applications adding new concepts or attributes of the concepts. In case you create a completely new strategy you have to add it to the author's course list following a link to http://localhost:8080/aha/Config and selecting "Change author option". You have also to add necessary variables to be initialized in the registration form aha/courseName/Registration.html and aha/courseName/index.html. "courseName" is the name of the application to which you want to apply the strategy.

QUESTIONNAIRE *Please answer the following questions, by deleting the answer not appropriate (in Yes/No questions) and by filling in the blanks and tables where required.*

1. Name:
2. Student ID:
3. email:
4. Study direction:
5. Did you hear before this course about learning styles? Yes/No
6. Are you familiar with XML? Yes/No
7. Did you understand how to work with the Graph Author? Yes/No
   - Did you experience any problems with editing the existing applications? Yes/No
   - Did you understand how the application of the strategies works in the delivery engine? Yes/No
   - Are you satisfied with the resulting presentation? Yes/No
8. Do the questionnaire at: http://www.engr.ncsu.edu/learningstyles/ilsweb.html and paste here the result (e.g., as GIF file):
9. **Instructional Strategies**: Compare the questionnaire results above (point 8) with your preferences as stated in table 1.

Strategy name	**Are the preferred settings as selected by you the same as in the questionnaire?(Yes/No)**
VerbalizerVersusImager	
ActivistVerusReflector	
GlobalVersusAnalytic	
Your own strategy? (replace w. name)	

Table A.3: Your own preferences

   If you replied somewhere no, try and explain in your own words why you think the difference appears?

10. **Instructional Meta- strategies (Monitoring strategies)**: Compare the induced preference (as filled-in in Table 2) with the questionnaire results in point 8. Fill in the table below:

    If you replied somewhere no, try and explain in your own words why you think the difference appears?

Strategy name	System induced preference same as questionnaire? (Yes/No)
TextVersusImagePreference	
BFVersusDFpreference	
Your own strategy? (replace w. name)	

Table A.4: Your own preferences

11. **Combination of Instructional Strategies**: Is there a difference between applying more strategies in different order (e.g., AudioVersusVideoPreference followed by VerbalizerVersusImager or vice-versa)? Yes/No

    If you replied yes, please explain in your own words what the differences are?

12. **Creation of Instructional (Meta-)Strategies**: you made some changes to the strategies at point 12 (in the task list). Did you get the result that you expected? Yes/No

    - If you answered no, please explain in your own words what you tried to do and didn't work.
    - Can you think of more strategies that you would like to apply but are not able to express using XML adaptation language? Please list them together with their description, in the table below (add more rows if necessary):

Strategy name	System induced preference same as questionnaire? (Yes/No)

Table A.5: Your own preferences

13. Do you find that application of different instructional/monitoring strategies for educational adaptive hypermedia useful? Yes/No Please explain your answer?

14. The work on the assignment was: pleasant/unpleasant easy/difficult

15. Do you have any suggestions for improvement or comments you wish to make?

Thank you for your cooperation!

Appendix B

Questionnaire of LAG-XLS

All data in this questionnaire is only used for testing and improving the LAG-XLS language, as well as the LAG-XLS tool evaluated, as well as statistical and qualitative data in publications on the language and the tool. No personal information will be used for any other purpose than the ones mentioned above. No personal information will be shown in any publications based on these evaluations, except when specifically allowed by the provider of that information (e.g., in special thanks, etc.). If you have any concerns or reservations about the usage of this data, do not hesitate to contact Natalia Stash at n.v.stash@tue.nl.

Please, answer the following questions:

1. Name
2. Could you describe your experience in the field of learning styles? What are the learning styles dimensions you have been mostly dealing with?
3. Have you had any experience with adaptive hypermedia before? Yes/No
4. Do you consider the idea of application of learning styles in online courses a useful endeavor? Yes/No

   Why?
5. Did you understand how application of learning styles in AHA! works (how the domain model is created, how the strategies are applied in the Graph Author and how their application influences the course presentation)?

   Yes/Yes, but had some difficulties/No

   In case "Yes, but had some difficulties" and "No" could you explain what these difficulties and problems were:
6. Did you understand what types of strategies can be expressed in our learning styles adaptation language LAG-XLS? Yes/No

   If "No", could you explain what was not clear:

7. Please, indicate:

   a. whether you were satisfied with the effect of the predefined instructional strategies for a number of learning styles and functionality of monitoring strategies in LAG-XLS,

   b. whether you would like to modify them and

   c. whether it is possible to express necessary modifications in our LAG-XLS language.

Predefined strategies	a. Level of satisfaction	b. Modifications, if necessary	c. Can modifications be expressed in LAG-XLS?
Instructional strategies:			
Activist vs. reflector	Very low/Low/Medium/High/Very high		Yes/No/Not sure
Imager (visualizer) vs. verbalizer	Very low/Low/Medium/High/Very high		Yes/No/Not sure
Global vs. analytic	Very low/Low/Medium/High/Very high		Yes/No/Not sure
Field-dependent vs. field-independent	Very low/Low/Medium/High/Very high		Yes/No/Not sure
Monitoring strategies:			
Image vs. text preference	Very low/Low/Medium/High/Very high		Yes/No/Not sure
Breadth-first vs. depth-first navigation preference	Very low/Low/Medium/High/Very high		Yes/No/Not sure

8. In case you have been using (or prefer using) other learning styles dimensions than described above, do you consider that some of the predefined strategies implemented in LAG-XLS can be simply reused (with or without your modifications) to represent these learning styles? (Some learning styles are different in definition, but similar in the treatment applied to students which use that style. Is that the case with any of the strategies that we have defined and that can be used — maybe with a small modification — for a learning style strategy that you are familiar with?)

9. In case you want to create a completely new strategy from scratch, could you give a short description for it and an indication about whether you consider it can/cannot be expressed in LAG-XLS?

10. Have you ever had experience with other systems attempting to provide adaptation to learning styles? Yes/No

    If yes, could you describe your experience?

11. What is possibly lacking in our approach?

12. What would you see as the main strengths of the system?

13. What would you see as the main weaknesses?
14. Do you have other remarks or comments?

Date

Thank you for your cooperation

Bibliography

IMS Learning Design Best Practice and Implementation Guide, Version 1.0 Final Specification. Technical report, 2003a.

IMS Simple Sequencing Information and Behavior Model, Version 1.0 Final Specification. Technical report, 2003b.

P. Ágh and M. Bieliková. Considering Human Memory Aspects to Adapting in Educational Hypermedia. In *Proc. of Workshop on Individual Differences at the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eindhoven, The Netherlands*, pages 107–114, 2004.

F. Albrecht, N. Koch, and T. Tiller. SmexWeb: an Adaptive Web-Based Hypermedia Teaching System. *Journal of Interactive Learning Research, Special Issue on Intelligent Systems/Tools in Training and Lifelong Learning*, pages 367–388, 2000.

Monique Ansems. AHA! Zichtbaarheid van Fragmenten (in Dutch). Technical report, Eindhoven University of Technology, 2002.

L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. INTRIGUE: Personalized Recommendation of Tourist Attractions for Desktop and Handset Devices. *Applied Artificial Intelligence Journal, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries*, 17(8-9):687–714, 2003.

B. Ash. Identifying Learning Styles and Matching Strategies for Teaching and Learning. Massachusetts: ERIC Document Reproduction Service (Education Resources Information Center, ED 270142, 1986.

F. A. Asnicar and C. Tasso. ifWeb: a Prototype of User-Model Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web. In *Proc. of the Workshop on Adaptive Systems and User Modeling on the World Wide Web at the 6th international concference on User Modeling (UM 97), Chia Laguna, Sardinia, Italy*, pages 3–11, 1997.

D. P. Ausubel, J. D. Novak, and H. Hanesia. *Educational Psychology: a Cognitive View.* New York: Holt, Rinehart, & Winston, 1978.

N. Bajraktarevic, W. Hall, and P. Fullick. Incorporating Learning Styles in Hypermedia Environment: Empirical Evaluation. In *Proc. of AH2003: Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, Budapest, Hungary*, pages 41–52, 2003.

E. Baralis and J. Widom. An Algebraic Approach to Static Analysis of Active Database Rules. *ACM Transactions on Database Systems*, 20(1):269–332, 2000.

A. J. Berlanga, F. J. Garca, and J. Carabias. Authoring Adaptive Learning Designs Using IMS LD. In *Proc. of the Adaptive Hypermedia and Adaptive Web-Based Systems conference (AH2006), Dublin, Ireland*, pages 31–40, 2006.

J. Bissell, S. White, and G. Zivin. Sensory Modalities in Children's Learning. *Psychology and educational practice*, page 130155, 1971.

G. Booch, I. Jacobson, and J. Rumbaugh. *The Unified Modeling Language User Guide.* The Addison-Wesley Object Technology Series. 1998.

T. Boyle and M. Encarnacao. MetaDoc: an Adaptive Hypertext Reading System. *User Modelling and User Adapted Interaction*, 4(1):1–19, 1994.

E. Brown, C. Stewart, and T. Brailsford. Adapting for Visual and Verbal Learning Styles in AEH. In *IEEE Proc. of International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade, The Netherlands*, pages 1145–1146, 2006.

M. N. Brumby. Consistent Differences in Cognitive Styles Shown for Qualitative Biological Problem-Solving. *British Journal of Educational Psychology*, 52:244257, 1982.

P. Brusilovsky. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11: 87–110, 2001.

P. Brusilovsky. *Authoring Tools for Advanced Technology Learning Environment*, chapter Developing adaptive educational hypermedia systems: From design models to authoring tools, pages 377–409. Dordrecht: Kluwer Academic Publishers, 2003.

P. Brusilovsky. Intelligent Tutor, Environment and Manual for Introductory Programming. *Educational and Training Technology International*, 29(1):26–34, 1992.

P. Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

P. Brusilovsky and L. Pesin. ISIS-Tutor: an Adaptive Hypertext Learning Environment. In *Proc. of JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering, Pereslavl-Zalesski, Russia*, pages 83–87, 1994.

P. Brusilovsky and C. Karagiannidisand D. Sampson. Layered Evaluation of Adaptive Learning Systems. *International Journal of Continuing Engineering Education and Life-long Learning*, 14(4/5):402–421, 2004.

P. Brusilovsky, E. Schwarz, and G. Weber. ELM-ART: an Intelligent Tutoring System on World Wide Web. In *Proc. of the 3rd international conference on Intelligent Tutoring Systems ITS'96, Montreal, Canada*, pages 261–269, 1996.

P. Brusilovsky, J. Eklund, and E. Schwarz. Web-Based Education for All: a Tool for Development Adaptive Courseware. *Computer Networks and ISDN Systems*, 30(1-7): 291–300, 1998.

V. Bush. As We May Think. *The Atlantic Monthly*, 176(1):101–108, 1945.

L. Calvi and A. Cristea. Towards Generic Adaptive Systems: Analysis of a Case Study. In *Proc. of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer Verlag, LNCS 2347*, pages 77–87, 2002.

L. Calvi and P. De Bra. Using Dynamic Hypertext to Create Multi-Purpose Textbooks. In *Proc. of the AACE Ed-Media'97 Conference, Calgary, Canada*, pages 130–135, 1997.

J. Canavan. Personalized E-Learning Through Learning Style Aware Adaptive Systems. Master's thesis, University of Dublin, Ireland, 2004.

M. Cannataro and A. Pugliese. *Web Dynamics. Adapting to Change in Content, Size, Topology and Use*, chapter A Survey of Architectures for Adaptive Hypermedia, pages 357–386. Springer-Verlag, Berlin, Germany, 2004.

M. Cannataro, A. Cuzzocrea, C. Mastroianni, R. Ortale, and A. Pugliese. Modeling Adaptive Hypermedia with an Object-Oriented Approach and XML. In *Proc. of the Second International Workshop on Web Dynamics (WebDyn'02), Honolulu, Hawaii*, May 2002.

R. M. Carro, E. Pulido, and P. Rodriguez. Designing Adaptive Web-Based Courses with TANGOW. In *Proc. of the 7th International Conference on Computers in Education (ICCE'99), Chiba, Japan*, volume 2, pages 697–704, November 1999.

C. A. Carver, R. A. Howard, and E. Lavelle. Enhancing Student Learning by Incorporating Learning Styles into Adaptive Hypermedia. In *Proc. of ED-MEDIA'96 World Conference on Educational Multimedia and Hypermedia, Boston, USA*, pages 118–123, 1996.

R. G. G. Cattell, D. K. Barry, and D. Bartels. *The Object Database Standard: ODMG 2.0*. Morgan-Kaufmann Series in Data Management Systems. 1997.

S. Ceri, P. Fraternali, and A. Bongio. Web Modeling Language (WebML): a Modeling Language for Designing Web Sites. *Computer Networks*, 33(1-6):137–157, June 2000.

S. Ceri, F. Daniel, M. Matera, and F. M. Facca. Model-driven Development of Context-aware Web Applications. *ACM Transactions on Internet Technology*, 7(1), 2007.

P. P. Chen. The Entity-relationship Model, Towards a Unified View of Data. *ACM-Transactions on Database Systems*, 1(1):9–36, 1976.

S. Chen and R. Macredie. Cognitive Styles and Hypermedia Navigation: Development of a Learning Model. *Journal of the American Society for Information Science and Technology*, 53(1):3–15, 2002.

K. Cheverst, k. Mitchell, and N. Davies. The Role of Adaptive Hypermedia in a Context-Aware Tourist Guide. *Communications of the ACM*, 45(5):47–51, 2002.

A. Cini and J. Valdeni de Lima. Adaptivity Conditions Evaluation for the User of Hypermedia Presentations Built with AHA! In *Proc. of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer Verlag, LNCS 2347*, pages 490–493, 2002.

J. A. Clarke. Cognitive Style and Computer Assisted Learning Problems and a Possible Solution. *Association for Learning Technology Journal (ALT-J)*, 1(1):47–59, 1993.

F. Coffield, D. Moseley, E. Hall, and K. Ecclestone. *Learning Styles and Pedagody in Post-16 Learning: a Systematic and Critical Review.* Learning and Skills research centre, http://www.lsda.org.uk/files/pdf/1543.pdf, 2004.

J. Collins, J. Greer, V. Kumar, and G. McCalla. Inspectable User Models for Just-Intime Workplace Training. In *Proc. of the Sixth International Conference on User Modeling, UM97*, pages 327–337, 1997.

J. Conklin. Hypertext: an Introduction and Survey. *IEEE Computer*, 20(9):17–41, 1987.

O. Conlan, V. Wade, C. Bruen, and M. Gardan. Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In *Proc. of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain*, pages 100–111, 2002.

A. Cristea and L. Calvi. The Three Layers of Adaptation Granularity. In *Proc. of the User Modeling conference*, pages 4–14, 2003.

A. Cristea and P. Cristea. Evaluation of Adaptive Hypermedia Authoring Patterns During a Socrates Programme Class. *Advanced Technology of Learning*, 1(2), 2004.

A. Cristea and A. De Mooij. LAOS: Layered WWW AHS Authoring Model and Its Corresponding Algebraic Operators. In *Proc. of the 14th international World Wide Web conference (WWW2003), Alternate Education track, Budapest, Hungary*, 2003.

A. I. Cristea, D. Smits, and P. De Bra. Writing MOT, Reading AHA! - Converting Between an Authoring and a Delivery System for Adaptive Educational Hypermedia. In *Proc. of A3EH Workshop at the AIED 2005 conference, Amsterdam, The Netherlands*, pages 107–114, 2005a.

A.I. Cristea and A. de Mooij. Adaptive Course Authoring: My Online Teacher. In *Proc. of the International Conference on Telecommunications, ICT'03, Papeete, French Polynesia*, pages 1762–1769, 2003.

A.I. Cristea, C. Stewart, T. Brailsford, and P. Cristea. Evaluation of Interoperability of Adaptive Hypermedia Systems: Testing the MOT to WHURLE Conversion in a Classroom Setting. In *Proc. of A3EH Workshop at the AIED 2005 conference, Amsterdam, The Netherlands*, pages 107–114, 2005b.

L. Curry. An Organization of Learning Styles Theory and Constructs. In *Paper presented at the Annual Meeting of the American Educational Research Association, Montreal, Quebec*, 1983.

D. Dagger, O. Conlan, and V. Wade. An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources. In *Proc. of World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education E-Learn 2003, Phoenix, Arizona, USA*, pages 49–56, 2003a.

D. Dagger, V. Wade, and O. Conlan. Towards "Anytime, Anywhere" Learning: The Role and Realization of Dynamic Terminal Personalization in Adaptive eLearning. In *Proc. of World Conference on Edicational Multimedia, Hypermedia and Telecommunications Ed-Media 2003, Honolulu, Hawaii, USA*, pages 32–35, 2003b.

H. L. Daniels. *Interaction of Cognitive Style and Learner Control of Presentation Mode in a Hypermedia Environment*. PhD thesis, Blacksburg, Virginia, 1996.

J. K. Davis. *Educational Implications of Field Dependence-Independence. In S. Wapner and J. Demick (Eds.), Field dependence-independence: cognitive style across the lifespan.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1991.

P. De Bra and L. Calvi. Creating Adaptive Hyperdocuments for and on the Web. In *Proc. of the AACE WebNet'97 Conference, Toronto, Canada*, pages 149–154, 1997.

P. De Bra and L. Calvi. AHA! an Open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia*, 4:115–139, 1998.

P. De Bra and J. P. Ruiter. AHA! Adaptive Hypermedia for All. In *Proc. of the AACE WebNet 2000 conference*, pages 262–268, 2001.

P. De Bra and N. Stash. Multimedia Adaptation Using AHA! In *Proc. of the ED-MEDIA 2004 Conference, Lugano, Switzerland*, pages 563–570, 2004.

P. De Bra, G. J. Houben, and H. Wu. AHAM: a Dexter-Based Reference Model for Adaptive Hypermedia. In *Proc. of the ACM Conference on Hypertext and Hypermedia, Darmstadt, Germany*, pages 147–156, 1999.

P. De Bra, A. Aerts, G. J. Houben, and H. Wu. Making General-Purpose Adaptive Hypermedia Work. In *Proc. of the AACE WebNet 2000 conference*, pages 117–123, 2000.

P. De Bra, A. Aerts, and B. Rousseau. Concept Relationship Types for AHA! 2.0. In *Proc. of the AACE ELearn'2002 conference*, pages 1386–1389, 2002.

P. De Bra, A. Aerts, D. Smits, and N. Stash. AHA! Version 2.0, More Adaptation Flexibility for Authors. In *Proc. of the AACE ELearn'2002 conference*, pages 240–246, 2002a.

P. De Bra, A. Aerts, D. Smits, and N. Stash. AHA! The Next Generation. In *Proc. of the ACM conference on Hypertext and Hypermedia*, 2002b.

P. De Bra, A. Aerts, D. Smits, and N. Stash. AHA! Meets AHAM. In *Proc. of the 2nd international conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer LNCS 2347*, pages 381–384, 2002c.

P. De Bra, A. Aerts, B. Berden, and B. De Lange. Escape from the Tyranny of the Textbook: Adaptive Object Inclusion in AHA! In *Proc. of the AACE ELearn 2003 Conference, Phoenix, Arizona,*, pages 65–71, 2003a.

P. De Bra, N. Stash, and B. De Lange. AHA! Adding Adaptive Behavior to Websites. In *Proc. of the NLUUG conference, Ede, The Netherlands*, 2003b.

P. De Bra, L. Aroyo, and V. Chepegin. The Next Big Thing: Adaptive Web-Based Systems. *Journal of Digital Information*, 5, 2004a.

P. De Bra, N. Stash, and D. Smits. Creating Adaptive Textbooks with AHA! In *Proc. of the AACE ELearn'2004 Conference, Washington DC, USA*, pages 2588–2593, 2004b.

P. De Bra, N. Stash, and D. Smits. Creating Adaptive Applications with AHA!, Tutorial for AHA! Version 3.0. Technical report, Eindhoven University of Technology, August 2004c. Tutorial at the AH2004 Conference, Eindhoven, The Netherlands.

P. De Bra, D. Smits, and N. Stash. The Design of AHA! In *Proc. of the ACM Conference on Hypertext and Hypermedia, Odense, Denmark*, page 133, 2006.

H. Dieterich, U. Malinowski, T. Kuhme, and M. Schneider-Hufschmidt. State of the Art in Adaptive User Interfaces. *Computing and Informatics*, 18:357–366, 1993.

R. Dunn. *The Dunn and Dunn Learning Style Model and Its Theoretical Cornerstone. Synthesis of the Dunn and Dunn Learning Styles Model Research: Who, What, When, Where and So What the Dunn and Dunn Learning Styles Model and Its Theoretical Cornerstone, 1-6.* New York: St John's University, 2003a.

R. Dunn. *Epilogue: So What? Synthesis of the Dunn and Dunn Learning Styles Model Research: Who, What, When, Where and So What the Dunn and Dunn Learning Styles Model and Its Theoretical Cornerstone.* New York: St John's University, 2003b.

R. Dunn and S. A. Griggs. *Learning Styles: a Quiet Revolution in American Secondary Schools*. Reston, VA: National Association of Secondary School Principals, 1988.

e-learning Consortium. Making Sense of Learning Specifications & Standards: a Decision Maker's Guide to Their Adoption. Technical report, Masie Center, 2002.

J. Eklund and P. Brusilovsky. InterBook: an Adaptive Tutoring System. *UniServe Science News*, 12:8–13, 1999.

J. Eklund, P. Brusilovsky, and E. Schwarz. Adaptive Textbooks on the WWW. In *Proc. of AUSWEB97The third Australian conference on the World Wide Web, Queensland, Australia*, page 186192, 1997.

M. Encarnacao. *Concept and Realization of Intelligent Support in Interactive Graphics Applications*. PhD thesis, University of Tübingen, Wilhelm-Schickard-Institute for Computer Science, Germany, 1997.

M. Encarnacao and A. Stork. An Integrated Approach to User-Centered Interface Adaptation. Technical report, University of Tübingen.

N. Entwistle. *Styles of Learning and Teaching: an Integrated Outline of Educational Psychology - for Students, Teachers and Lecturers*. Chichester: John Wiley, 1981.

N. J. Entwistle. *Improving Teaching Through Research on Student Learning*. University teaching: international perspectives. New York: Garland, 1998.

R. M. Felder. Reaching the Second Tier: Learning and Teaching Styles in College Science Education. *College Science Teaching*, 23(5):286–290, 1993.

R. M. Felder and E. R. Henriques. Learning and Teaching Styles in Foreign and Second Language Education. *Foreign Language Annals*, 28(1):2131, 1995.

R. M. Felder and L. K. Silverman. Learning and Teaching Styles in Engineering Education. *Journal of Engineering Education*, 78(7):674–681, 1988.

R. M. Felder and B. A. Soloman. Index of Learning Styles Questionnaire.

R. M. Felder and J. E. Spurlin. Applications, Reliability, and Validity of the Index of Learning Styles. *Intl. Journal of Engineering Education*, 21, 2005.

J. Fink and A. Kobsa. User Modeling in Personalized City Tours. *Artificial Intelligence Review Journal*, 18(1):33–74, 2002.

N. Ford. Learning Styles and Strategies of Postgraduate Students. *British Journal of Educational Psychology*, 16:65–79, 1985.

N. Ford. Levels and Types of Mediation in Instructional Systems: an Individual Differences Approach. *International Journal of Human-Computer Studies*, 43:241259, 1995.

N. Ford and S. Y. Chen. Matching/Mismatching Revisited: an Empirical Study of Learning and Teaching Styles. *British Journal of Educational Psychology*, 32(1):5–22, 2001.

R. D. Freedman and S. A. Stumpf. Learning Style Theory: Less Than Meets the Eye. *Academy of Management Review*, 6:297–299, 1980.

R. W. Gardner, P. S. Holzman, G. S. Klein, H. B. Linton, and D. P. Spence. Cognitive Control. A Study of Individual Consistencies in Cognitive Behavior. *Psychological Issues. Monograph 4. N. Y.*, 1, 1959.

S. Garlatti, S. Iksal, and P. Kervella. Adaptive On-line Information System by Means of a Task Model and Spatial Views. In *2nd Workshop on Adaptive Systems and User Modeling on World Wide Web at WWW1999 conference, Toronto, Canada*, pages 59–66, 1999.

A. F. Grasha. Learning Styles: the Journey from Greenwich Observatory (1796) to the College Classroom (1984). *Improving College and University Teaching*, 32(1):46–53, 1984.

A. F. Gregorc. Learning/Teaching Styles: Potent Forces Behind Them. *Educational Leadership*, 36:234–237, 1979.

A. F. Gregorc. Style as a Symptom: a Phenomenological Perspective. *Theory into Practice*, 23(1):5155, 1984.

E. L. Grigorenko and R. J. Sternberg. *Thinking Styles*. International Handbook of Personality and Intelligence. New York: Plenum Press, 1995.

M. Grigoriadou, K. Papanikolaou, H. Kornilakis, and G. Magoulas. INSPIRE: an Intelligent System for Personalized Instruction in a Remote Environment. In *Proc. of 3rd Workshop on Adaptive Hypertext and Hypermedia, Sonthofen, Germany*, pages 13–24, 2001.

F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. In *Proc. of the NIST Hypertext Standardization Workshop, Gaithersburg, MD, USA*, pages 95–133, January 1990.

F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30–39, 1994.

S. E. Hartman, J. Hylton, and R. F. Sanders. The Influence of Hemispheric Dominance on Scores of the Myers-Briggs Type Indicator. *Educational and Psychological Measurement*, 57(2):440449, 1997.

J. Hayes and C. W. Allison. Matching Learning Style and Instructional Strategy: an Application of the Person-Environment Paradigm. *Perc. and Motor Skills*, 76:63–79, 1993.

J. G. Hedberg and S. E. McNamara. The HumanTechnology Interface: Designing for Open and Distance Learning. *Educational Media International*, 26(2):73–81, 1989.

P. L. Heineman. Cognitive Versus Learning Style. 1995.

N. Henze and W. Nejdl. Student Modeling in an Active Learning Environment Using Bayesian Networks. Technical report, University of Hannover, Germany, 1998.

N. Henze and W. Nejdl. Bayesian Modeling for Adaptive Hypermedia Systems. Technical report, University of Hannover, Germany, 1999.

M. A. Holodnaya. *Cognitive Styles: About the Nature of Individual Mind (in Russian)*. Per Se, Moscow, 2002.

P. Honey and A. Mumford. *The Learning Styles Helper's Guide*. Maidenhead: Peter Honey Publications Ltd, 2000.

P. Honey and A. Mumford. *Using Your Learning Styles*. Maidenhead: Peter Honey Publications, 1986.

P. Honey and A. Mumford. *The Manual Learning Styles*. Maidenhead: Peter Honey Publications, 1992.

D. H. Jonassen and B. Grabowski. *Individual Differences and Instruction*. New York: Allen & Bacon, 1993.

S. Jones. Cognitive Learning Styles: Does Awareness Help? A Review of Selected Literature. *Language Awareness*, 2(4):195207, 1993.

C. G. Jung. *Analytical Psychology: Its Theory and Practice the Tavistock Lectures*. New York: Random House, 1968.

I. Juvina. *Development of a Cognitive Model for Navigating on the Web*. PhD thesis, Utrecht University, The Netherlands, ISBN-10: 90-393-4382-9, ISBN-13: 978-90-393-4382-1, 2006.

J. Kagan. Reflection-Impulsivity: the Generality and Dynamics of Conceptual Tempo. *J. of Abnorm. Psychology*, 71:17–24, 1966.

J. W. Keefe. Learning Style: an Overview. *NASSP's Student learning styles: Diagnosing and prescribing programs, Reston, VA: National Association of Secondary School Principals*, pages 1–17, 1979.

R. M. Keller, S. R. Wolfe, J. R. Chen, J. L. Rabinowitz, and N. Mathe. A Booking Service for Organizing and Sharing URLs. In *Proc. of the 6th international conference on World Wide Web, Santa Clara, California, USA*, pages 1103–1114, 1997.

A. Kobsa. Personalized Hypermedia and International Privacy. *Communications of the ACM*, 45(5):64–67, 2002.

A. Kobsa, J. Koenemann, and W. Pohl. Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. *The Knowledge Engineering Review*, 16: 111–155, 2001.

N. Koch and M. Wirsing. The Munich Reference Model for Adaptive Hypermedia Applications. In *Proc. of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems (AH'02), Malaga, Spain*, pages 213–222, May 2002.

D. A. Kolb. *Facilitator's Guide to Learning*. Boston: Hay/McBer, 2000.

D. A. Kolb. Experiential Learning Theory and the Learning Style Inventory: a Reply to Freedman and Stumpf. *Academy of Management Review*, 6(2):289–296, 1981.

D. A. Kolb. *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, New Jersey: Prentice Hall, 1984.

D. A. Kolb. *The Kolb Learning Style Inventory, Version 3*. Boston: Hay Group, 1999.

M. Kravcik, M. Specht, and R. Oppermann. Evaluation of WINDS Authoring Environment. In *Proc. of the Adaptive Hypermedia conference (AH2004), Eindhoven, The Netherlands*, 2004.

R. Kulhavy and W. Stock. Feedback in Written Instruction: The Place of Response Certitude. *Educational Psychology Review*, 1(4):279–308, 1989.

M. Laroussi and M. Benahmed. Providing an Adaptive Learning Through the Web Case of CAMELEON: Computer Aided MEdium for LEarning on Networks. In *Proc. of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering, Göteborg, Sweden*, pages 411–416, 1998.

W. S. Li, Q. Vu, E. Chang, D. Agrawal, K. Hirata, S. Mukherjea, Y. L. Wu, C. Bufi, C. C. K. Chang, Y. Hara, R. Ito, Y. Kimura, K. Shimazu, and Y. Saito. PowerBookmarks: a System for Personalizable Web Information Organization, Sharing and Management. *ACM SIGMOD*, 28(2):565–567, 1999.

Y. Liu and D. Ginther. Cognitive Styles and Distance Education. *Online Journal of Distance Learning Administration*, 2(3), 1999.

J. McCrone. "Right Brain" or "Left Brain" Myth or Reality? *The New Scientist, RBI Limited*, 2000.

K. Meng and D. Patty. Field-dependence and Contextual Organizers. *Journal of Educational Research*, 84(3):183189, 1991.

M. D. Merrill. *Instructional Strategies and Learning Styles: Which Takes Precedence?* Prentice Hall, trends and issues in instructional technology edition, 2000.

S. Messick. *Individuality in Learning.* San Francisco: Jossey-Bass, 1976.

S. Messick. The Nature of Cognitive Styles: Problems and Promise in Educational Practice. *Educational Psychologist*, 19(2):5974, 1984.

D. P. Mitchell. *Learning Style: a Critical Analysis of the Concept and Its Assessment.* London: Kogan Page, 1994.

T. J. F. Mitchell, S. Y. Chen, and R. D. Macredie. Adapting Hypermedia to Cognitive Styles: Is It Necessary? In *Proc. of AH2004: Workshop on Individual Differences in Adaptive Hypermedia*, pages 70–80, 2004.

A. Mitrovic. SQL-Tutor: a Preliminary Report. Technical report, University of Canterbury, New Zealand, 1997.

D. Mladenic. Machine Learning Used by Personal Webwatcher. In *Proc. of the Workshop on Machine Learning and Intelligent Agents (ACAI 99), Chania, Crete*, 1999.

W. Nejdl and M. Wolpers. KBS Hyperbook - a Data-Driven Information System on the Web. In *Proc. of the 8th World Wide Web conference, Toronto, Canada*, 1999.

J. D. Novak and A. J. Canas. The Theory Underlying Concept Maps and How to Construct Them. Technical report, Florida Institute for Human and Machine Cognition, USA, 2006.

K. Papanikolaou, M. Grigoriadou, H. Kornilakis, and G. D. Magoulas. Personalising the Interaction in a Web-Based Educational Hypermedia System: the Case of INSPIRE. *User Modeling and User-Adapted Interaction*, 13(3):213–267, 2003.

P. Paredes and P. Rodrigues. Considering Sensing-intuitive Dimension to Exposition-exemplification in Adaptive Sequencing. In *Proc. of the Adaptive Hypermedia (AH2002) Conference, Malaga, Spain*, page 556559, 2002.

G. Pask. Styles and Strategies of Learning. *British Journal of Educational Psychology*, 46: 128–148, 1976.

G. Pask and B. C. E. Scott. Learning Strategies and Individual Competence. *International Journal of Man-Machine Studies*, 4:217–253, 1972.

F. Paterno and C. Mancini. Designing Web Interfaces Adaptable to Different Types of Use. In *Proc. of the Workshop Museums and the Web*, http://www.archimuse.com/mw99/papers/paterno/paterno.html, 1999.

N. J. C. Primus. A Generic Framework for Evaluating Adaptive Educational Hypermedia Authoring Systems. Evaluating MOT, AHA! and WHURLE to Recommend on the Development of AEH Authoring Systems. Master's thesis, University of Twente, March 2005.

J. Reiff. At-risk Middle Level Students or Field Dependent Learners. *Clearing House*, 69 (4):231234, 1996.

E. Rich. Users Are Individuals: Individualizing User Models. *International Journal of Human-Computer Studies*, 51:323–338, 1999.

A. Richardson. Verbalizer-Visualizer: a Cognitive Style Dimension. *Journal of Mental Imagery*, 1(1):109–126, 1997.

R. Riding. *Cognitive Styles Analysis CSA Administration*. Birmingham: Learning and Training Technology, 1991a.

R. Riding. *Cognitive Styles Analysis Users' Manual*. Birmingham: Learning and Training Technology, 1991b.

R. Riding. *Cognitive Styles Analysis - CSA Administration*. Birmingham: Learning and Training Technology, 1991c.

R. Riding. *Cognitive Styles Analysis Research Applications*. Birmingham: Learning and Training Technology, 1991d.

R. Riding and G. Douglas. The Effect of Cognitive Style and Mode of Presentation on Learning Performance. *British Journal of Educational Psychology*, 63:297–307, 1993.

R. Riding and S. Rayner. *Cognitive Styles and Learning Strategies: Understanding Style Differences in Learning Behaviour*. London: David Fulton Publishers Ltd, 1998.

R. J. Riding. On the Nature of Cognitive Style. *Educational Psychology*, 17(1-2):29–49, 1997.

R. J. Riding and I. Cheema. Cognitive Styles - an Overview and Integration. *Educational Psychology*, 11(3-4):193–215, 1991.

R. J. Riding and S. Rayner. The Information Superhighway and Individualized Learning. *Educational Psychology*, 15(4):365–378, 1995.

C. Romero, P. De Bra, S. Ventura, and C. de Castro. Using Knowledge Levels with AHA! for Discovering Interesting Relationships. In *Proc. of the AACE ELearn'2002 Conference*, 2002.

C. Romero, S. Ventura, C. Hervás, and P. De Bra. Extending AHA!: Adding Levels, Data Mining, Tests and SCORM to AHA! In *Proc. of the Fifth International Conference on Human Systems Learning: Who is in control? Marrakech, Morocco*, pages 21–40, 2005.

C. Romero, S. Ventura, C. Hervás, and P. De Bra. An Authoring Tool for Building Both Mobile Adaptable Tests and Web-Based Adaptive or Classic Tests. In *Proc. of the Adaptive Hypermedia and Adaptive Web-Based Systems conference (AH2006), Dublin, Ireland*, pages 203–212, 2006.

S. M. Rundle and R. Dunn. *The Guide to Individual Excellence: a Self Directed Guide to Learning and Performance Solutions.* New York: Performance Concepts International, 2000.

W. Smith, S. Sekar, and K. Townsend. The Impact of Surface and Reflective Teaching and Learning on Student Academic Success. In In M. Valcke, Learning styles: reliability D. Gombeir, and validity, editors, *Proc. of the 7th Annual European Learning Styles Information Network Conference, 2628 June, Ghent*, pages 407–418, 2002.

I. L. Sonnier. Hemisphericity: a Key to Understanding Individual Differences Among Teachers and Learners. *Journal of Instructional Psychology*, 18(1):1722, 1991.

M. Specht and R. Opperman. ACE-Adaptive Courseware Environment. *The New Review of Hypermeida and Multimedia*, 4:141161, 1998.

R. W. Sperry. The Great Cerebral Commissure. *Scientific American*, 210(1):4252, 1964.

S. P. Springer and G. Deutsch. *Left Brain, Right Brain.* New York: Freeman, 1989.

N. Stash and P. De Bra. Building Adaptive Presentations with AHA! 2.0. In *Proc. of the PEG conference, Saint-Petersburg, Russia*, 2003.

N. Stash and P. De Bra. Incorporating Cognitive Styles in AHA! (the Adaptive Hypermedia Architecture). In *Proc. of the IASTED International Conference Web-Based Education, Innsbruck, Austria*, pages 378–383, 2004.

N. Stash, A. Cristea, and P. De Bra. Authoring of Learning Styles in Adaptive Hypermedia: Problems and Solutions. In *Proc. of the WWW 2004 Conference, New York, NY, USA*, pages 114–123, 2004.

N. Stash, A. Cristea, and P. De Bra. Explicit Intelligence in Adaptive Hypermedia: Generic Adaptation Languages for Learning Preferences and Styles. In *Proc. of HT2005 CIAH Workshop, Salzburg, Austria*, 2005.

N. Stash, A. Cristea, and P. De Bra. Learning Styles Adaptation Language for Adaptive Hypermedia. In *Proc. of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems AH'2006, Dublin, Ireland*, pages 323–327, 2006a.

N. Stash, A. Cristea, and P. De Bra. Adaptation to Learning Styles in E-Learning: Approach Evaluation. In *Proc. of E-Learn 2006 Conference, Honolulu, Hawaii*, 2006b.

M. Stern and P. Woolf. Adaptive Content in an Online Lecture System. In *Proc. of the International Conference on Adaptive Hypermedia and Adaptive Web-based systems, Trento, Italy*, pages 291–300, 2000.

R. J. Sternberg. *Thinking Styles.* Cambridge: Cambridge University Press, 1999.

C. Stewart, A. Moore A. Cristea, T. Brailsford, and H. Ashman. Authoring and Delivering Adaptive Courseware. In *Proc. of the 2nd workshop on Adaptive and Adaptable Educational Hypermedia at the Adaptive Hypermedia conference (AH2004), Eindhoven, The Netherlands*, 2004.

C. Stewart, A.I. Cristea, T. Brailsford, and H. Ashman. "Authoring Once, Delivering Many": Creating Reusable Adaptive Courseware. In *Proc. of the Web-Based Education Conference WBE'05, Grindelwald, Switzerland*, 2005.

M. Tennant. *Psychology and Adult Learning.* London: Routledge, 1998.

J. K. Di Tiberio. *Education, Learning Styles and Cognitive Styles. In AL Hammer (ed.) MBTI applications: a decade of research on the Myers-Briggs Type Indicator.* Palo Alto, CA: Consulting Psychologists Press, 1996.

C. Tinajero and M. F. Paramo. Field Dependence-independence and Academic Achievement: a Re-examination of Their Relationship. *British Journal of Educational Psychology*, 67:199–212, 1997.

C. Tinajero, M. F. Paramo, F. Cadaveira, and S. Rodriguez-Holguin. Field Dependence-Independence and Brain Organization: the Confluence of Two Different Ways of Describing General Forms of Cognitive Functioning? A Theoretical Review. *Perceptual and Motor Skills*, 77:787802, 1993.

E. Triantafillou, A. Pomportsis, and E. Georgiadou. AES-CS: Adaptive Educational System Based on Cognitive Styles. In *Proc. of the Adaptive Hypermedia (AH2002) Workshop, Malaga, Spain*, pages 10–20, 2002.

J. van Ossenbruggen, J. Geurts, F. Cornelissen, L. Rutledge, and L. Hardman. Towards Second and Third Generation Web-Based Multimedia. In *Proc. of the Tenth International World Wide Web Conference, ACM Press, Hong Kong*, pages 479–488, 2001.

J. Vassileva. A Task-Centered Approach for User Modeling in a Hypermedia Office Documentation System. *User Modeling and User Adapted Interaction Journal*, 6(2-3):185–223, 1996.

J. Vassileva, J. Greer, G. McCalla, R. Deters, D. Zapata, C. Mudgal, and S. Grant. A Multi-Agent Design for a Peer-Help Environment. In *Proc. of the 10th international conference on Artificial Intelligence in Education (AIED2001), San Antonio, Texas, USA*, pages 38–45, 2001.

J. D. Vermunt. Metacognitive, Cognitive and Affective Aspects of Learning Styles and Strategies: a Phenomenographic Analysis. *Higher Education*, 31:25–50, 1996.

D. M. Wardell and J. R. Royce. Towards a Multi-Factor Theory of Styles and Their Relationship to Cognition and Affect. *J. of Personality*, 46(3):474–505, 1978.

H. A. Witkin and D. R. Goodenough. *Cognitive Styles: Essence and Origins*. New York: International Universities Press, 1981.

H. A. Witkin, Ph. K. Oltman, E. Raskin, and S. Karp. *A Manual for the Embedded Figures Tests*. Cognitive Psychology Press, Inc., 1971.

H. A. Witkin, C. A. Moore, D. R. Goodenough, and P. W. Cox. Field-dependent and Field-independent Cognitive Styles and Their Educational Implications. *Review of Educational Research*, 47:164, 1977.

C. Wolf. iWeaver: Towards an Interactive Web-Based Adaptive Learning Environment to Address Individual Learning Styles. *European Journal of Open and Distance Learning*, 2002.

H. Wu. *A Reference Architecture for Adaptive Hypermedia Applications*. PhD thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2, 2002.

H. Wu, E. De Kort, and P. De Bra. Design Issues for General-Purpose Adaptive Hypermedia Systems. In *Proc. of the ACM Conference on Hypertext and Hypermedia (Hypertext'01), Aarhus, Denmark*, pages 141–150, August 2001.

G. S. Yoon. The Effects of Instructional Control, Cognitive Style and Prior Knowledge on Learning of Computer-Assisted Instruction. *Journal of Educational Technology Systems*, 22(4):357–370, 1993.

M. S. Zywno. A Contribution to Validation of Score Meaning for Felder-Soloman's Index of Learning Styles. In *Proc. of the 2003 American Society for Engineering Education Annual Conference and Exposition*, 2003.

List of Figures

List of Tables

Index

Summary

This dissertation brings together two disciplines — **Adaptive Hypermedia**, a *computer science* topic and **Cognitive/Learning Styles**, a research topic in *cognitive psychology.*

Adaptive hypermedia (AH) addresses the usability problems users often encounter in traditional hypermedia applications, such as navigation and comprehension problems. These problems are caused by the rich link structure of the applications. Adaptive hypermedia systems (AHS) can support the user in navigation by limiting the browsing space; they can guide the user through the information towards relevant pages so that (s)he can reach them more easily and quickly. Furthermore AHS allow to improve comprehension of the content by presenting the most relevant information on a page to the user and hiding information that is not relevant. In the field of adaptive hypermedia these forms of adaptation are referred to as *adaptive navigation support* (or *link adaptation*) and *adaptive presentation* (or *content adaptation*). For each user, an AHS constructs and maintains a user model that represents various characteristics of the user and his/her environment and applies this information for providing adaptation. Information about the user is gathered by the system by observing his/her behavior which is most often browsing.

Typically AHS are developed for a specific application area and cannot be reused in another context. The most popular area for adaptive hypermedia research is adaptive educational hypermedia systems (AEHS). The goal of the user (learner) is to learn all the learning material or a reasonably large part of it. The key feature in this area is the user's knowledge of the subject area being studied. Other learner features the AEHS may take into account are background, hyperspace experience, preferences and interests. These properties tend to relate to the application domain. Less attention has been paid in the adaptive hypermedia field to the fact that people have different approaches to learning, namely that individuals perceive and process information in very different ways. These differences are referred to as cognitive/learning styles and are studied by Cognitive Psychology.

The aim of this dissertation is twofold. First, it aims at developing a general-purpose system that targets various application domains and is capable of performing adaptation to many different aspects of users. Second, it pays special attention to the application of learning styles (LS) in adaptive hypermedia. It thus bridges the gap between the disciplines of adaptive hypermedia and cognitive psychology.

Chapter 1 introduces the research questions and approaches and the outline of this dissertation.

Chapter 2 gives an overview of adaptive hypermedia systems and existing models for adaptive hypermedia. From these models, we select the Adaptive Hypermedia Application Model (AHAM) as a basis for creating a general-purpose tool, and formulate a number of design requirements.

Chapter 3 describes the design and implementation of a general-purpose tool. Rather than start from scratch (or from the AHAM model) we have developed the general-purpose AHA! system (version 3.0) based on an existing educational adaptive hypermedia engine. This was used in the course "Hypermedia Structures and Systems", known as 2L670 and later 2L690, at the Eindhoven University of Technology (TU/e). The chapter presents the overall AHA! 3.0 architecture and different AHA! sub-models with respect to AHAM. It discusses whether and how AHA! implements the design requirements formulated in chapter 2 and pays attention to the issue of authoring adaptive applications in AHA!. We show that building a general-purpose tool is a time consuming and difficult process. Though we can already consider AHA! as a general-purpose system, since it can be used for creating applications targeting various domains and can provide adaptation to various aspects of users, not all of the design requirements are currently satisfied by AHA!.

Chapter 4 presents general information about the field of learning styles and puts stress on those that, according to psychology researchers, have potential implications for pedagogy.

Chapter 5 discusses the suggestions from psychological and computer science research for possible adaptation of instructional design to a number of learning styles. It presents an overview of existing AHS that provide learning styles adaptation. By analyzing existing implementations we identify a number of issues we wish to improve in our approach. The overview shows that only a few systems can provide certain types of adaptation independently of the learning style model. In most systems, the choice of learning style and the corresponding instructional strategy are predefined by the designers of the systems. Furthermore in most systems, the LS are assessed through psychological questionnaires and psychometric tests. The disadvantage of using questionnaires is that they are time consuming, the results are not always reliable and valid and moreover not all characteristics they measure are stable and invariable across different subject domains.

We present our approach to incorporating learning styles in AHA! in chapter 6. We consider that the task of choosing the learning style model and defining the learning styles strategies should be left to the authors of adaptive applications rather than the developers of the system. We define an adaptation language for learning styles, called LAG-XLS. Authors can create their strategies in this language and decide which strategies to apply to which application. LAG-XLS not only allows the definition of instructional strategies as identified in chapter 5, but also the monitoring strategies for inferring preferences corresponding to particular learning styles. These inference strategies can serve as an alternative solution in cases when the time consuming process of filling in long psychological questionnaires is undesirable. The chapter compares our approach with the approaches adopted in other systems providing LS adaptation. It also shows the results of evaluation of this approach with students from the TU/e and the results of approach validation performed by a number of LS experts.

Samenvatting

Dit proefschrift brengt twee disciplines samen — **Adaptieve Hypermedia**, een onderwerp uit de *informatica* en **Cognitieve/Leer- Stijlen**, een onderzoeksonderwerp in *cognitieve psychologie*.

Adaptieve Hypermedia (AH) richt zich op de bruikbaarheidsproblematiek die de gebruikers van traditionele hypermedia toepassingen vaak hebben, zoals navigatie en begrip problemen. Deze problemen worden veroorzaakt door de veelheid aan verbindingen in de toepassingen. Adaptieve Hypermedia Systemen (AHS) kunnen de gebruiker bij de navigatie assisteren door bij het doorbladeren het aantal verbindingen te beperken; zij kunnen de gebruiker naar relevante pagina's begeleiden zodat hij/zij hen makkelijker en sneller kan vinden. Verder helpen AHS om de inhoud beter te begrijpen door de meest relevante informatie op de pagina aan de gebruiker te tonen en informatie die niet relevant is te verbergen. Op het gebied van Adaptieve Hypermedia worden deze vormen van adaptatie als *adaptieve navigatiehulp* (of *link adaptatie*) en *adaptieve presentatie* (of *inhoud adaptatie*) genoemd. Voor elke gebruiker construeert en onderhoudt een AHS een gebruikersmodel, wat een representatie is van de diverse kenmerken van de gebruiker en zijn omgeving. Deze informatie wordt door AHS voor adaptatie gebruikt. De informatie over de gebruiker wordt door het systeem verzameld door het blader-gedrag te observeren.

Typisch worden AHS voor een specifiek toepassingsgebied ontwikkeld waardoor ze niet kunnen herbruikt worden in een andere context. Het meest populair gebied voor adaptieve hypermedia onderzoek betreft adaptieve educatieve hypermedia systemen (AEHS). Het doel van de gebruiker (leerling) is normaliter om al het studiemateriaal of een redelijk groot deel ervan te leren. De belangrijkste eigenschap op dit gebied is de kennis die de gebruiker heeft van het onderwerp dat wordt bestudeerd. De andere eigenschappen van de student waarmee een AEHS rekening kan houden zijn achtergrond, hyperspace ervaring, voorkeur en interesses. Dit zijn eigenschappen die vaak nog steeds met het toepassingsdomein gerelateerd zijn. Minder aandacht is besteed in het adaptieve hypermedia veld aan het feit dat de mensen verschillende benaderingen hebben van het leren, namelijk dat de individuen informatie op zeer verschillende manieren waarnemen en verwerken. Deze verschillen worden cognitieve/leer- stijlen genoemd en in de Cognitieve Psychologie bestudeerd.

Het doel van dit proefschrift is tweeledig. Ten eerste beoogt het de ontwikkeling van een systeem voor algemeen gebruik dat zich op diverse toepassingsdomeinen richt en kan worden aangepast aan de vele verschillende aspecten van de gebruikers. Ten tweede besteedt het speciale aandacht aan de toepassing van de verschillende leerstijlen (LS) in adaptieve hypermedia. Het overbrugt zo de hiaat tussen de disciplines van adaptieve hypermedia en cognitieve psychologie.

Hoofdstuk 1 introduceert de onderzoeksvragen en benaderingen en geeft een overzicht van dit proefschrift.

Hoofdstuk 2 geeft een overzicht van de adaptieve hypermedia systemen en de bestaande modellen voor adaptieve hypermedia. In dit hoofdstuk hebben wij ervoor gekozen om het Adaptive Hypermedia Application Model (AHAM) toe te passen om een algemeen bruikbaar gereedschap te creëren en wij hebben de ontwerpeisen voor zo'n tool (gebaseerd op AHS en AHAM beschrijvingen) geïnventariseerd.

Hoofdstuk 3 toont het ontwerp en de implementatie van een hulpmiddel voor algemeen gebruik. In plaats van geheel op nieuw te beginnen hebben wij het AHA! systeem (versie 3.0) voor algemeen gebruik ontwikkeld op basis van de educatieve adaptieve hypermedia engine die wordt toegepast in de cursus "Hypermedia Structures and Systems" bekend als 2L670 en later als 2L690, aan de Technische Universiteit Eindhoven (TU/e). Het hoofdstuk beschrijft de algemene architectuur van AHA! 3.0 en de verschillende AHA! submodellen met betrekking tot AHAM. Het bespreekt of en hoe AHA! aan de ontwerpeisen voldoet die in hoofdstuk 2 worden geformuleerd en besteedt aandacht aan de kwestie van auteurs of ontwerpers van adaptieve toepassingen in AHA!. Dit hoofdstuk toont aan dat de ontwikkeling van een hulpmiddel voor algemeen gebruik een tijdrovend en moeilijk proces is. Hoewel wij AHA! reeds kunnen beschouwen als een systeem voor algemeen gebruik, aangezien het voor het creëren van toepassingen kan worden gebruikt met als doelstelling diverse domeinen en aanpassing aan diverse aspecten van de gebruikers te verstrekken, zijn nog niet alle ontwerpeisen vervuld in AHA!.

Hoofdstuk 4 geeft algemene informatie over het gebied van leerstijlen en benadrukt die leerstijlen die volgens psychologie onderzoekers mogelijke implicaties voor pedagogie hebben.

Hoofdstuk 5 bespreekt de suggesties van psychologisch- en informatica-onderzoek voor mogelijke aanpassingen van het instructie ontwerp aan een aantal leerstijlen. Het presenteert een overzicht van de bestaande AHS voor LS aanpassing. Door te analyseren wat de andere onderzoekers gedaan hebben, om daarna dezelfde functionaliteit ook in ons systeem te implementeren hebben wij ook een aantal kwesties geïdentificeerd die wij in onze aanpak proberen te verbeteren. Het overzicht toont aan dat er slechts een paar systemen zijn die bepaalde soorten aanpassing van adaptatie aan kunnen onafhankelijk van het desbetreffende leerstijl model. In de meeste systemen worden de keus van de leerstijl en de overeenkomstige educatieve strategie vooraf bepaald door de ontwerpers van de systemen. Verder worden LS in de meeste systemen beoordeeld door de psychologische vragenlijsten en psychometrische tests. Het nadeel van vragenlijsten is dat zij vrij lang zijn en de resultaten niet altijd betrouwbaar en geldig zijn; bovendien zijn niet alle kenmerken die zij meten stabiel en onveranderlijk tussen de verschillende onderworpen.

Wij stellen onze benadering van het opnemen van adaptatie aan leerstijlen in AHA! voor in hoofdstuk 6. Wij zijn van mening dat het kiezen van een leerstijl model en het creëren van LS strategieën de taak van de auteur van adaptieve toepassingen is eerder dan van de ontwikkelaars van het systeem. Wij hebben de adaptatie-taal voor leerstijlen, LAG-XLS, gedefineerd. De auteurs kunnen hun strategieën in deze taal realiseren en bepalen welke strategieën van toepassing zijn op bepaalde applicaties. Met LAG-XLS is het niet alleen mogelijk om de educatieve strategieën te definiëren zoals diegene die in hoofdstuk 5 bepaald zijn, maar ook de controle strategieën om de voorkeuren van de gebruikers te bepalen. Deze controle strategieën kunnen als alternatieve oplossing dienen in gevallen waar het tijdrovende proces om lange psychologische vragenlijsten in te vullen ongewenst is. Het hoofdstuk vergelijkt onze aanpak met de benaderingen die in andere systemen worden gevolgd om adaptatie voor leerstijlen toe te passen. Het toont ook de resultaten van evaluatie van deze benadering met de studenten van de TU/e en de resultaten van een enquête die onder een aantal leerstijl deskundigen werden uitgevoerd.

Curriculum Vitae

Natalia Stash was born on February 6, 1976, in Leningrad (now Saint-Petersburg), Soviet Union (now Russian Federation).

She received her engineer diploma with specialization in Control and Information Technology in Engineering Systems in 1999 at Saint-Petersburg State Electrotechnical University (LETI) in Saint-Petersburg, Russia. The title of the diploma is "Development of the distance learning server for studying of artificial intelligence". In the same year she started a PhD course at the same university but left in 2001 to the Netherlands for an assistant researcher position in the AHA! project (Adaptive Hypermedia for All) at the Department of Mathematics and Computer Science in Eindhoven University of Technology (TU/e). In 2003 her contract was extended for writing a PhD thesis. Since 2005 she has a position of a scientific programmer at the same department in the CHIP project (Cultural Heritage Information Personalization) — a collaboration project between the Rijksmuseum in Amsterdam, TU/e and Telematica Institute (Enschede) funded by the CATCH program of the Dutch Science Foundation NWO.

Her research interests include adaptive Web-based systems, semantic Web technologies, learning styles, e-culture applications. Natalia Stash can be reached at n.v.stash@tue.nl.

SIKS Dissertatiereeks

====
1998
====

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations
within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

====
1999
====

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling;
Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven
Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems

1999-7 David Spelt (UT)
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent
Mechanism for Discrete Reallocation.

====
2000
====

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie;
een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations,
Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

====
2001
====

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with
Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,

Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a multiagent modeling and simulation language
for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management:
The Role of Mental Models in Business Systems Design

====
2002
====

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments
inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology;
Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative
E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and
Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

====
2003
====

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction
between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to
Digital Media Warehouses

2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
Learning Search Decisions

====
2004
====

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar
abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale
politiële gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining

2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)
Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

====
2005
====

2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM))
AI techniques for the game of Go

2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM)
Adaptive Game AI

2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems

2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments

2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry

2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU)
Usability of XML Query Languages

2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

====
2006
====

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU)
Building Web Service Ontologies

2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines

2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling

2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion

2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment, and the arts

2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change

2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN)
Aptness on the Web

2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlovic (UT)
Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

====
2007
====

2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU)
Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System