Web-based educational hypermedia

P. De Bra
Department of Computer Science
Eindhoven University of Technology, The Netherlands

Abstract

The Web has revolutionized the way information is delivered to people throughout the world. It did not take long for learning material to be delivered through the Web, by using electronic textbooks. The use of hypertext links gives the learner a lot of freedom to decide on an order in which to study the material. This leads to problems in understanding the textbook, which can be solved by using methods and techniques. In this chapter we describe how the field of educational hypermedia benefits from and . We also show that the information gathered about the learners and their learning process can be used to improve the quality of the electronic textbooks.

1 Introduction

For a long time the use of in education was limited because of the need for specialized hardware/software platforms for bringing that hypermedia to the end-user. Some readers may remember the Plato system [1], featuring personal and group notes, threaded discussions, hyperlinks, interactive elements and games, etc. Its use was restricted to institutes equipped with special terminals, connected to large mainframe computers running the Plato environment. A more affordable but also unsuccessful attempt at bringing hypermedia and interactivity to the public was the introduction of CD-Interactive by Philips and Sony (in 1986). Many other (hypermedia) environments suitable for delivering interactive learning material have been developed and never become popular. Then, in 1989, Tim Berners Lee started developing the ideas and software for the World Wide Web. The first implementations of Web Servers (from CERN and NCSA, much later evolved into the most popular Apache server) and Browsers (NCSA Mosaic for X-Windows being the first popular one, later evolved into Netscape and now Mozilla Firefox) initially only allowed for a fairly primitive form of hypermedia: pages with (untyped) links, and possibly some embedded images. Soon a primitive form of servers-side dynamic content generation was added through CGI scripts (for Common Gateway Interface). No matter how primitive, World Wide Web immediately offered an irresistible package: a simple server running on inexpensive Unix workstations, enabling everyone to set up their own server, a simple completely text-based page format (HTML) that everyone could master very easily, and a simple browser running on everyone's workstation. This, combined with computers being connected through Internet and a universal way to address pages on any machine anywhere on the Web turned everyone into a potential publisher and gave everyone access to all the information published on the Web.

Around 1993 people started publishing course material on the Web rather than in paper documents (or on proprietary systems in proprietary document formats). At the Eindhoven University of Technology we started the first on-line course on the subject of Hypermedia, using the Web as the hypermedia platform for delivering the course. (This course is currently still available (in updated form) as http://wwwis.win.tue.nl/2L690/.) Our early experience with this course [2] made us aware of a number of difficulties learners experience when using a course text that is delivered in hypermedia form:

- The first problem we encountered was . A course text delivered through hypermedia need not have a strictly hierarchical structure of chapters, sections, subsections and paragraphs. In our hypermedia course there is a list of major topics, but quite a lot of pages can be categorized as belonging to different topics and are also reachable through the introductory pages about different topics. As a result a typical hierarchical table-of-content-like navigation aid indicating the position of the current page in the structure, as found on many websites, would not be appropriate. (The current page often belongs in different places in the structure at once.) Not only was it difficult for learners to know "where" in the course text they were, it was also difficult for them to know how much they had already studied and how much more there was still to be read, and where that additional material could be found.
- The second problem was . Giving learners access to all the material at once, through a (clickable) list of topics, meant that the learners could just pick any topic and start studying it. The unnumbered list of topics apparently did not convey a message of desired reading order, in the way the sequential order of chapters and pages in a book does. It is hard, if not impossible, to write a course text in such a way that every page, or at least every chapter, can be read without having first studied some other part of the course.
- The third problem was of the learner's performance. We added a multiple-choice test to our hypermedia course, to assess the learner's knowledge after studying all the material. Since the reading (study) order was completely free there was no clear place in the hypermedia structure to place tests about partial knowledge, like at the end of each chapter (in a textbook). Students who considered themselves ready to take the final test and failed had no

clear way of knowing how to remedy that problem. Intermediate tests would reveal much earlier on whether they were studying the material in sufficient detail or not.

There are essentially only two ways to solve the above mentioned problems: create and present a clearly hierarchical structure with chapter and section numbers to indicate a preferred reading order, or provide some form of “” for users who wish to follow unforseen paths through the learning material. In the past the emphasis has often been on fairly strong guidance, in “Intelligent Tutoring Systems”, like ELM-ART [3, 4] and SQL-Tutor [5, 6] (among many others). In this chapter we will describe how methods and tools can be used to combine the desire for as much navigational freedom as possible, with the need to study certain concepts before building other knowledge upon them. In Section 2 we will have a look at the different methods and techniques that are used in adaptive hypermedia in general, and adaptive learning environments in particular, and indicate their purpose. In Section 3 we will sketch the overall architecture of adaptive systems using a reference model. In Section 4 we discuss the AHA! system developed at the Eindhoven University of Technology, and show different ways in which on-line course texts can be created with or for AHA!. We devote Section 5 on the issue of evaluating the learner’s knowledge and using that as input for further adaptation. In Section 6 we discuss the use of adaptation to serve learners with different cognitive styles. We conclude with some advice and outlook into the future of on-line adaptive educational hypermedia.

2 Adaptive (educational) hypermedia

In [7, 8] Peter Brusilovsky has presented an overview of the field of adaptive hypermedia. Rather than summarizing these overviews we concentrate on the use of adaptation in an educational environment. There are two main reasons for using adaptation: to avoid problems that occur when reading material out of the intended order, and to better serve the individual differences between users. The latter issue is discussed in Section 6. In the current section we concentrate on using adaptation to facilitate studying a course that offers a lot of navigational freedom (to pick topics to study at will).

In every course text there are a lot of cross-references between different chapters/sections. In a normal paper textbook the author knows whether such a reference is a forward or a backward reference, and thus whether the reference is to a concept not yet known resp. already known. In an on-line course with navigational freedom the author cannot know whether a reference is a forward or backward reference. However, it isn’t difficult to track a user’s path through the course text, and thus for the system to know whether for this user at this time a reference is a forward or backward reference. So if an author creates two versions of the reference the system can choose and present the appropriate one. This leads to the first important form of adaptation: or as Brusilovsky [7] defines it: .

There are essentially three cases where content adaptation because of a reference to another concept is useful. (See also [7].)

- When a reference is made to a concept the learner does not yet know, and of which at least some understanding is needed a short *prerequisite explanation* can be inserted. This lets the learner continue with the chosen subject, rather than requiring a jump to the prerequisite concept to study that in detail first.
- Sometimes the current concept can be elaborated upon further in case the related concept is already known, or when the knowledge level of the learner is already high. For these "expert" users an *additional explanation* can be given that is beyond the level of the average learner (at the time of visiting the current page).
- Sometimes an interesting comparison is possible with another concept, but only if that other concept is already known. Such a *comparative explanation* between the concepts can automatically be shown on the page of the second of the two concepts studied by the learner, whichever of the two is second in the chosen reading order.

When used with care the learner may not be aware that content adaptation is being performed. If content adaptation happens frequently and obviously the learners may become uneasy, thinking that they are "missing out" on some of the information in the course.

The differences in knowledge, caused by choosing different paths through the course text can be compensated for to some extent, but sometimes it is necessary to guide the users in a certain direction or to keep users away from some learning material they are really not ready for (meaning that a short prerequisite explanation is not sufficient to prepare the learner for the rest of the description of the topic). The reading order can be influenced by manipulating the hypertext links. This brings us to the second form of adaptation: *link adaptation*, Brusilovsky [7] defines it: *adaptive navigation support*.

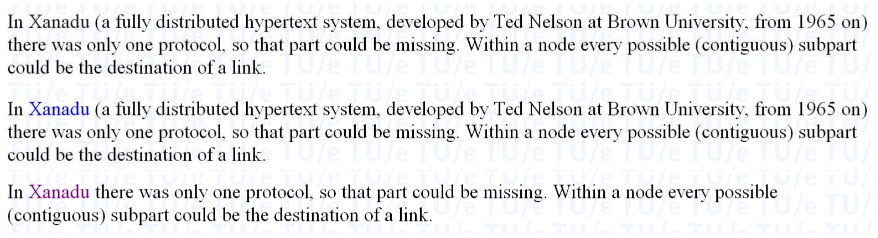
Link adaptation can be done in several ways, because links have a *position* on the page a *presentation* through a link anchor and a *destination*.

- That the *position* of links has an influence on the user's behavior is best seen in the presentation of results by search engines. People are used to lists having a meaning of ranking from best to worst. This perceived meaning can be used to sort lists of links so that the most appropriate links for the current learner, in his current state of mind, are placed at the top. Brusilovsky [7, 8] calls this *adaptive link sorting*. Besides link sorting, links can also be made "available" through a system of menus and submenus. A commonly used technique is to show a list of chapters, and the list of sections of the "current" chapter. By having only the sections of one chapter listed at any one time the user is suggested to only select sections from that one chapter and not jump around between sections in different chapters. Conditionally presenting (or hiding) links is also called adaptive *link removal*.
- Links are presented through a *link anchor*. This is a word or phrase or perhaps an icon, displayed in a way that hints at it being a link. On the Web links are typically represented by anchors that appear in blue and that are underlined, and images or icons that are link anchors typically have a blue border. In modern browsers the shape of the cursor changes when it is moved over a

link anchor. (It usually becomes a “hand”.) There are different ways in which the user can be guided towards or away from certain links by changing the presentation of the link anchor. Virtually all Web browsers change the color of the link anchor after the link is visited. (The standard color scheme is that blue link anchors become purple.) However, using style sheets it is possible to change the appearance of link anchors in other ways, and it is also possible to add icons to indicate a special meaning of a link. The AHA! system [9] has a default link adaptation technique of *link hiding*. Link anchors can be blue, purple or black, and link anchors are not underlined in AHA!, so the black links are effectively hidden. They look just like normal text. It is also possible to change this color scheme and use three visible colors, resulting in a technique of *link annotation*. Other systems, like ELM-ART [10] and Interbook [11] for instance, use icons to recommend for or against certain links (green and red balls).

- In almost all hypermedia literature there is a close connection between *links* and *link anchors*. Users expect a link anchor to always correspond to the same link, and hence to lead to the same link destination. However, when pages are generated dynamically (e.g. by an adaptive system) there is no technical reason why the same anchor cannot lead to a different destination, depending on the circumstances. A link can for instance lead to a simplified description of a topic or a detailed description depending on the learner’s progress. People are beginning to get used to (and accept) such unstable link anchors, because of “top stories”, “joke of the day” and other links that routinely lead to information that changes frequently. In a learning context, in [12, 13, 14], the use of an adaptive course (with adaptive tests) is described in which the outcome of a test would automatically lead to the start of a chapter at a beginner’s, intermediate or advanced level. It is almost like the automatic progression to higher levels in shootout or adventure games.

We conclude this section with a small example from the hypermedia course. Figure 1 shows three ways in which a short paragraph about the Xanadu system appears in a page about URLs. (See http://www.xanadu.net/XUarchive/ for an archive of notes about Xanadu.)



In Xanadu (a fully distributed hypertext system, developed by Ted Nelson at Brown University, from 1965 on) there was only one protocol, so that part could be missing. Within a node every possible (contiguous) subpart could be the destination of a link.

In Xanadu (a fully distributed hypertext system, developed by Ted Nelson at Brown University, from 1965 on) there was only one protocol, so that part could be missing. Within a node every possible (contiguous) subpart could be the destination of a link.

In Xanadu there was only one protocol, so that part could be missing. Within a node every possible (contiguous) subpart could be the destination of a link.

Figure 1: Content Adaptation in the Hypermedia Course.

In the first version a brief comparison is made between URLs and the addressing mechanism of Xanadu. Because the learner has not yet studied Xanadu a short note on Xanadu is added between parentheses. And although there is a whole page on Xanadu, the link to that page is hidden because the user has not yet started studying the whole chapter on the history of hypertext systems. The second version also shows the short note on Xanadu, but this time the user has already started the chapter on the history and hence the link to the page on Xanadu is recommended (shown in blue). In the third version the learner is reading this page after learning about Xanadu. The link is shown as visited (purple), meaning the learner can go to the Xanadu page but this is no longer recommended as the learner was there before. The short note about Xanadu is no longer shown, so as to not interrupt the sentence with that note. The learner already knows everything about Xanadu, so the note is no longer needed. This simple and small example shows how content adaptation and link adaptation can be combined to provide the "appropriate" description to every learner. What the example doesn't show is that the decision which adaptation *is* "appropriate" is often non-trivial. In most current adaptive systems an author has to decide under which circumstances which adaptation is made. This is laborious and error-prone. There is a potential benefit for some form of automated reasoning, based on data mining, machine learning or neural networks to relieve the author from this difficult task.

3 The AHAM reference architecture

In the period 1965-1990 a large number of hypertext systems were designed, built, and used, sometimes for many applications but often only for one or a few. Despite all the differences, researchers got together for a workshop (at the "Dexter Inn") and this resulted in the Dexter Model [15, 16], a formal model that captured the architecture of the existing (and some future) systems. The model consists of five layers covering different functional parts of a hypertext system. The most important part is the *storage layer* which describes the structure of nodes (pages) and links. This layer touches upon the anchoring and presentation-specifications layers. This model suits our purposes well because it separates the "hypermedia" part from the user-interface or browser part and also from the database or file system part (of how data are represented inside the components). In [17] we extended the Dexter model to describe the architecture of adaptive hypermedia systems. This resulted in the AHAM model, as shown in Figure 2.

The storage layer is extended to consist of three parts: the *domain model* which roughly corresponds to the "old" storage layer, representing the structure of the information, the *user model* which represents how the user relates to the information (e. g. in terms of knowledge), and the *teaching model*, later renamed to *adaptation model*, which represents the methods that are used to guide the learner through the information structure.

Without going into the details of the AHAM model, we will describe the main elements that show clearly why adaptive educational hypermedia systems fit so nicely in this model.

- When designing a course it is important to first list the concepts or topics

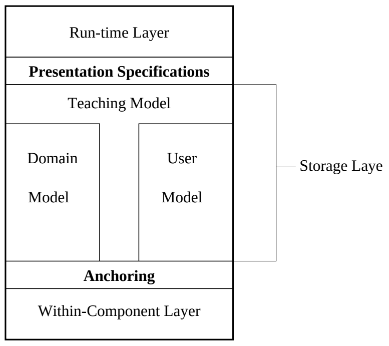


Figure 2: The AHAM model.

the course should cover. A hierarchy of topics and subtopics is a good start, because it can be translated into a structure of chapters, sections and paragraphs.

- The next thing to do is to determine dependencies between the concepts. In some subject areas such dependencies are pretty clear and rigid, like an argumentation line of lemmas and theorems in mathematics (where the proofs of one theorem depend on other theorems and lemmas), but in other areas it is also sometimes advisable to study certain concepts before certain other concepts. This leads to a structure of *prerequisite relationships*. Many other types of relationships can be thought of, but prerequisites are most common in educational material.
- The different techniques (like the *link annotation* and the *conditional inclusion of fragments*) mentioned in Section 2 can be used to translate the concept relationships into adaptive behavior. For each type of relationship (e. g. prerequisites) some *generic adaptation rules* can be defined. Likewise, the translation of the action of reading pages into knowledge increments for certain concepts (small page concepts and higher-level section and chapter concepts) is also defined through such adaptation rules. The rules together form what we call the “teaching model” or “adaptation model” in AHAM.

A key point in AHAM is that the adaptation rules are used to translate user actions into user model updates (as well as the presentation and adaptation of content). Also, the rules use user model information together with the action information in order to determine the required user model updates. A typical example of such an adaptation rule would be (expressed in plain English):
on access of page P, if P is recommended then set the knowledge value for P to 100
Expressed in this way the rules are so called *event-condition-action* rules, where *access of page P* is the event, *if P is recommended* is the condition checking the recommendation status of *P* in the user model, and *set the knowledge value for P to 100* is the action. A page access is just one type of event triggering such a rule. The evaluation of a test (e. g. a multiple-choice test) would be another triggering event, but user model updates are also triggers. When the knowledge of *P* is changed and

P belongs to a section *S* which belongs to chapter *C* then the knowledge update to *P* may trigger a rule that has an action to create a (smaller) knowledge update to *S* and that update may be the trigger for an (even smaller) knowledge update to *C*. Authors of learning material need not be familiar with these adaptation rules. They can expect their authoring environment to translate concept relationships into these adaptation rules automatically. The existence of a rule system in AHAM is important because the approach of events that conditionally trigger actions suggests a system architecture that is very suitable for creating logging information that can be mined in order to find interesting access, test and user model update patterns.

Many seemingly very different authoring and delivery platforms for adaptive electronic textbooks exist (e.g. KBS-HyperBook [18, 19], Interbook [11], Tangow [20], MOT [21] and AHA! [9]). However different, all systems consider concepts and concept relationships and have some type of intelligence to perform adaptation based on the concept relationships. In this sense all systems follow the general architecture of the AHAM reference model (but possibly with a different kind of rule or reasoning language and engine). In the next Section we are going to look at AHA!, a system that corresponds closely to AHAM, including the structure of the adaptation rule language.

4 A general-purpose adaptive web-based platform

AHA! is an Open Source Web server extension to add adaptation to applications such as on-line courses. In this chapter we concentrate on how AHA! can adapt a course to the individual learner. However, in order to perform that adaptation AHA! stores and updates a lot of information in the *user model* and optionally also maintains a complete *log* of all the user's actions, thus providing ample opportunity for data mining applications to analyze the users' behavior and detect potential issues or problems that are experienced by a significant number of learners [13, 12].

4.1 Overall AHA! architecture

Figure 3 shows the overall architecture of AHA!. The core is formed by the AHA! engine which is implemented using Java Servlets running on (and communicating with) the web-server. The information on the server consists of three parts we describe in detail below: a combined *domain and adaptation model* (DM/AM), corresponding to these models in AHAM, a *user model* (UM) which keeps track of the user's knowledge about the domain concepts, and the *local pages* which contain the content of the application or course. It is possible to include external pages (retrieved from other web servers) and they are (potentially) adapted in the same way as local pages. AHA! also contains authoring and management tools, explained in a later section.

- The basis for the adaptation is the information stored in the user model (UM). Apart from some information that is specific for the user and independent of the application or course the UM is an *overlay model*. For each

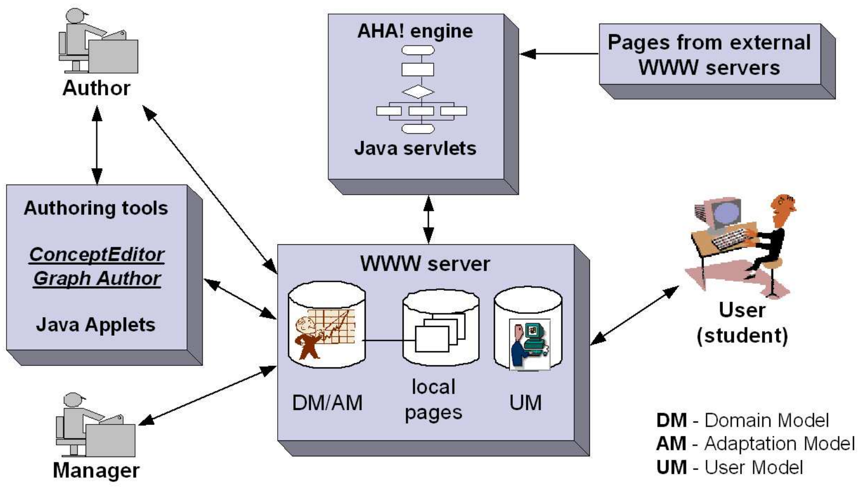


Figure 3: Overall AHA! Architecture.

concept covered by the course UM contains a number of attribute values. In AHA! each concept can have arbitrarily many (named) attributes, but typically there is at least a *knowledge* value and a *suitability*. The knowledge attributes of all concepts together give a detailed impression of the knowledge the learner has of the whole course. The suitability value represents whether the concept is recommended. In AHA! attributes can be persistent or volatile. When the suitability is volatile it means that whenever a link to a concept is shown the system must decide whether the prerequisites are satisfied (in order to adapt the link). When it is persistent the suitability is calculated when it changes and simply used when needed.

- The *conceptual structure* of an AHA! application consists of *concepts* that are optionally associated with (one or more *resources*). With each concept a number of attributes are associated, each with a number of adaptation rules. When a new user logs on the concept/attribute structure is copied into the UM. For each concept there is an attribute *access* with adaptation rules that are executed when the concept is accessed (because the user clicks on a link to that concept). The adaptation rules work exactly as described in the AHAM model in Section 3, including the propagation of rule execution and thus of knowledge updates from pages to sections to chapters.
- The (local) pages and fragments are represented in DM/AM as *resources*. There can be a fixed link between a concept and a resource, meaning that whenever the concept is accessed the resource that will be shown (possibly adapted) to the user is fixed. But it is also possible to let an adaptation rule (tied to a *show-ability* attribute) decide which resource to present (see below).

Figure 4 shows the adaptation performed by AHA! based on the adaptation rules and the UM. The figure is actually a slight simplification.

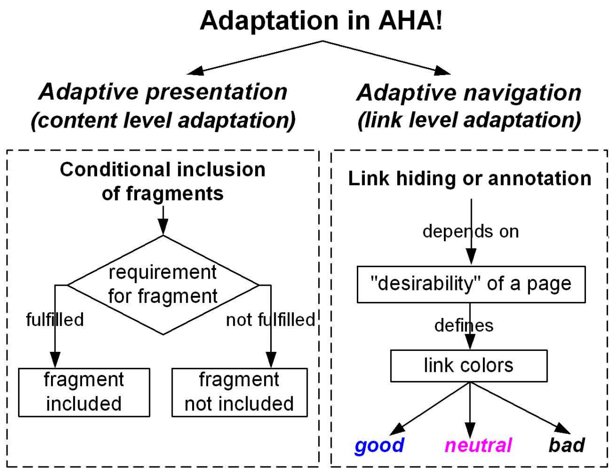


Figure 4: Content and Link Adaptation in AHA!.

- When a page (resource) is presented, it can be adapted in two ways: it can have *conditional fragments* or *conditional objects*. The difference is that a conditional fragment is embedded in the page (and so is the requirement to decide whether the fragment should be shown or not), and that a conditional object is just a reference to a concept. The engine decides which resource to include, so it is more than just a yes/no issue. Conditionally included fragments or objects can themselves contain more conditionally included fragments or objects.
- When a page contains a link the engine looks at the suitability to decide whether the destination of the link (a concept or the page the concept leads to) is recommended or "desired". The link is then presented accordingly. The standard default scheme in AHA! is to use a *blue* link for recommended and not previously visited link destinations, *purple* for recommended but already visited destinations, and *black* for non-recommended destinations. Optionally it is also possible to add icons to the link (and presented in front of or behind the link anchor). Also, the blue/purple/black color scheme can be changed (but the number of colors is fixed at three).

In AHA! the presentation of a course can be influenced in many more ways that are beyond the scope of this chapter. AHA! uses a *layout model* to define how concepts (of different types) are presented. A layout is an html frames structure, and apart from a frame that shows a page there can be frames that show part of a table of contents, concepts of which the knowledge is increased by reading the current page, prerequisite concepts (and their knowledge level), etc. Through the powerful layout model AHA! applications can be made to look very similar to applications in other adaptive educational hypermedia platforms. In [22] we made a first high-fidelity translation from Interbook [11] to AHA!, and we have since redesigned that translation to use high-level structures as described in the next section instead of the low level adaptation rules described above.

4.2 The AHA! authoring tools

AHA! comes with two main authoring tools (and a multiple-choice test editor [14]). For fast and easy authoring of the conceptual structure there is the *Graph Author*, in which the author can use high-level concept relationships such as prerequisites. The tool generates the adaptation rules for DM/AM automatically. For specialists or for fine-tuning the generated rules AHA! offers the *Concept Editor* (sometimes called *generatelist editor*). Figure 5 shows a screenshot of the Graph Author. This

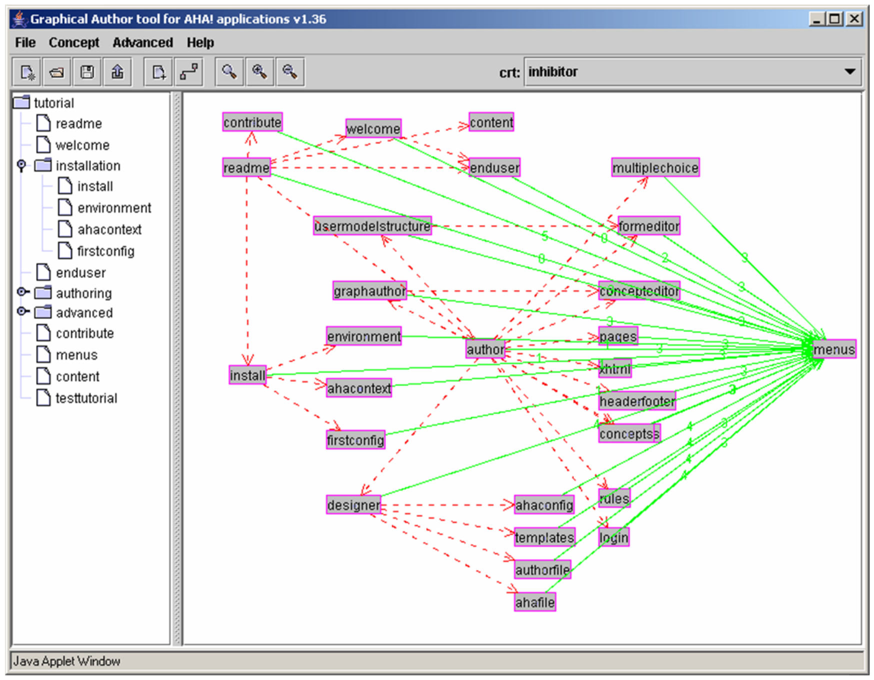


Figure 5: The Graph Author Tool.

figure shows the concept structure of an (old) AHA! tutorial. On the left the concept hierarchy is shown. This hierarchy is used to generate adaptation rules for propagating knowledge from pages to sections to chapters. On the right we see a graph of concept relationships. Each color and arrow style represents a different type of relationship. AHA! uses templates to translate the high-level relationships to the low level adaptation rules. By using high-level relationships it is much easier for an author to understand the structure that form the basis for the adaptation than by authoring adaptation rules directly. However, the AHA! engine only works with adaptation rules. If (through data mining for instance) one discovers that a certain adaptation rule does not produce an optimal adaptation effect it is not immediately clear which high-level relationship has to be changed.

5 Questions, quizzes and tasks

Adaptive educational hypermedia systems focus on optimizing the process of reading course material. The basic idea is that adaptation is based on the system's idea of the *knowledge* of the learner. However, the system does not have very reliable information to base that estimate of the user's knowledge on. Especially in a web-based system the server-side adaptation engine (like AHA!) only receives events when the user clicks on a link. Other user actions like scrolling (to the bottom of a page) and actual reading (that could be followed by tracking eye-movement) do not result in messages being sent to the server. Also, the time between page accesses is not a reliable measure for the actual reading time as users may be distracted and perform other tasks instead of reading.

It is clear that the system can get a more accurate impression of the user's knowledge by "probing" that knowledge through questions, quizzes, or the results of more elaborate tasks. Multiple-choice tests are the most popular, because they are easy to grade automatically. A slight variation, found in Interbook [11] for instance is a textfield in which a number, word or phrase must be entered (literally).

An interesting use of tests in an adaptive learning environment is in providing remedial content. After a test, the learner can be guided towards pages that need to be revisited because they seem not to be understood quite as well as the user model (based on reading alone) had indicated. But alternatively, tests can also be used to probe a user's knowledge before she starts reading, in order to decide which subjects can be skipped or need to be reviewed only briefly.

Another interesting part about adding tests or quizzes to an adaptive educational hypermedia environment is that the tests allow us to verify whether the knowledge level estimated through recording page accesses and the knowledge level measured through tests correspond. Monitoring this just in a single user is not meaningful because the learner may accidentally have missed something while reading, or misread something or forgotten a detail. When monitoring large numbers of users a pattern may emerge that indicates that learners systematically get certain questions wrong [13]. This does not immediately tell the course author what is wrong but it does point out which topic is not understood as well as it should.

6 Adapting to learning styles

In the previous sections we have only considered adaptation to the *knowledge level* of the learner. This level is mostly used in combination with *prerequisite relationships* in order to decide which content to show and how to annotate links. These adaptation methods and techniques treat every learner in exactly the same way. By considering individual differences in the way people prefer to learn the adaptation can be further improved. Some of this was tried in [23] and [24] for instance. An overview of possibilities and problems in using learning styles in adaptive hypermedia can be found in [25].

Roughly speaking there are two ways in which adaptation to learning styles is done: there are learning styles that indicate a preference for some *media type(s)* over others, and there are learning styles that influence the *order* in which learning material is studied.

Adaptation to media preferences can be easily implemented through the conditional inclusion of fragments or objects. When logging in for the first time the system can ask the learner whether she prefers presentations through text, images, video, audio or whichever media types are available. Alternatively the system can offer access to the different types of content and deduce the media preference from the choice the learner makes. ARTHUR [23] is an example of a system that uses media preferences. It distinguishes between the *visual-interactive*, *autidory-lecture* and *text* styles.

Adaptation of the *learning order* is more difficult and also more controversial. There are several classifications of learning styles, including Dunn and Dunn [26], Felder-Silverman [27], Kolb [28] and Honey and Mumford [29]. We show just two examples of how adaptation is possible to some of these categories of learning styles.

Kolb [28] distinguishes between *example-oriented* learners, or ”Reflectors” and *activity-oriented* learners, or ”Activists”. According to Kolb’s learning model, ”Reflectors” are people who tend to collect and analyze data before taking an action. ”Activists” are more motivated by experimentation and attracted by challenge. In order to adapt to these individual differences one can add an attribute to concepts to indicate whether they represent an example or an activity. On a page or menu that provides links to the pages that are available on a certain topic the example can be presented and recommended first to the reflectors and the activity can be presented and recommended first to the activists. This type of adaptation can be taken one step further because often there is not just examples and activities but for instance also theory or a definition or explanation. Some learners want to see examples first, then the theory, then do an activity, whereas others may want to see the theory first, then examples, then activity, and others... Adaptation to these types of learning styles only involves *sorting*, *conditional inclusion of fragments* and *link annotation*.

Learners will generally not know terms like “Reflector” and “Activist” (in their Kolb learning style meaning). The learning style can be probed by offering the learner the free choice between example, activity or explanation at first, and by observing a pattern in the choices she makes.

A pair of more complex types of to adapt to are the *field-dependent* (FD) and *field-independent* (FI) styles [30]. FI users follow an analytical approach. They pick a topic and study it in detail. FD users need to first see the global picture and ignore the details (until later). Also, FD learners are more likely to require externally defined goals and reinforcements, whereas FI ones tend to develop self-defined goals. Assuming that the “global” picture can be gathered by reading the top-level or introductory pages of each chapter of a course, we can start by considering the lower-level pages as non-recommended for FD learners. Optionally we may even wish to write special introductory pages for each topic, pages that assume the learner wishes to get the general picture and not yet start studying the details. In other words, for FD learners the system should (at least for the top levels) suggest (or maybe even enforce) a breadth-first navigation through the concept hierarchy.

FI learners can pick a chapter and start studying it right away without visiting the introductions of other chapters. So this learning style does not require adaptation. (FI users need not be forced to stay within a chapter they start for instance.)

Detecting FD versus FI is not trivial because the beginning of a breadth-first navigation pattern is not yet a clear sign of an FD learner and by the time the system would know for sure the learner is FD the breadth-first part of the navigation may well be over. An alternative is to use a questionnaire, but reliable questionnaires for determining learning styles are long and cumbersome.

If the cognitive style of users is known beforehand it is interesting to see whether using the adaptation suggested above actually improves the learning process and outcome, and/or whether users (always) follow the advice that corresponds to their learning style. We have not performed any user studies to confirm or deny the potential positive effect of performing adaptation to learning styles in the way we suggested above.

7 Conclusions

In this chapter we have briefly reviewed the area of web-based educational hypermedia, and in particular the use of *adaptive hypermedia* technology in this area. We have shown a reference architecture (AHAM) for adaptive hypermedia and given some details of the AHA! system that resembles this reference architecture most closely. We have shown that high-level authoring tools can shield course authors from the low-level adaptation rules that drive the adaptation engine. The adaptation is first and foremost driven by the conceptual structure of the learning material. We also looked into the use of learning styles to improve the learning process further by taking individual differences between users (independent of the topic of the learning material) into account.

Because web-based systems record access events, the updating of the user model that keeps track of the user’s knowledge and also the logging of the access events provide interesting data that can be used to analyze whether course material and tests are properly matched to each other and whether the adaptation helps to provide the appropriate learning material to the learner.

We have not covered all possible uses of adaptation in e-learning. Some researchers have concentrated on providing adaptation based on information about the user’s task [31] or other contextual aspects. We have also not covered the issue of *search* in adaptive educational hypermedia, which is a difficult topic because the adaptation makes it difficult to predict what exactly the content will be of search “hit” (a link to a concept), or more precisely, of the page that will appear when the learner clicks on that link.

References

[1] Wooley, D.R., Plato: The emergence of online community. *Matrix News, (online version at wwwthinkofitcom/plato/dwplatohtm)*, 1994.

[2] de Bra, P., Teaching hypertext and hypermedia through the web. *Journal of Universal Computer Science (JUCS)*, **2(12)**, 1996.

[3] Weber, G. & Specht, M., User modeling and adaptive navigation support in www-based tutoring systems. *Int. Conference on User Modeling*, pp. 289–300, 1997.

[4] Weber, G. & Brusilovsky, P., Elm-art: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education*, **12**, pp. 351–384, 2001.

[5] Mitrovic, A., Experiences in impelementing constraint-based modeling in sql-tutor. *Int. Conf. on Intelligent Tutoring Systems*, pp. 414–423, 1998.

[6] Mitrovic, A., Using evaluation to shape its design: Results and experiences with sql-tutor. *User Modeling and User-Adapted Interaction*, **12(2–3)**, pp. 243–279, 2002.

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

[8] Brusilovsky, P., Adaptive hypermedia. *User Modeling and User Adapted Interaction*, **11(1)**, 2001.

[9] de Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D. & Stash, N., Aha! the adaptive hypermedia architecture. *ACM Conference on Hypertext and Hypermedia*, Nottingham, UK, pp. 81–84, 2003.

[10] Brusilovsky, P., Schwarz, E. & Weber, G., Elm-art: An intelligent tutoring system on world wide web. *International Conference on Intelligent Tutoring Systems*, pp. 261–269, 1996.

[11] Brusilovsky, P., Eklund, J. & Schwarz, E., Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems (WWW’98 Conference)*, **30(1–7)**, 1998.

[12] Romero, C., Ventura, S. & de Bra, P., Knowledge discovery with genetic programming for providing feedback to courseware author. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, **14(5)**, pp. 425–464, 2004.

[13] Romero, C., Ventura, S., de Castro, C. & de Bra, P., Discovering prediction rules in aha! courses. *Lecture Notes in Artificial Intelligence*, **2702**, pp. 25–34, 2003.

[14] Romero, C., Martin-Palomo, S., de Bra, P. & Ventura, S., An authoring tool for web-based adaptive and classic tests. *AACE ELEARN’2003 Conference*, Phoenix, AZ, pp. 174–177, 2003.

[15] Halasz, F. & Schwartz, M., The dexter reference model. *NIST Hypertext Standardization Workshop*, pp. 95–133, 1990.

[16] Halasz, F. & Schwartz, M., The dexter hypertext reference model. *Communications of the ACM*, **37(2)**, pp. 30–39, 1994.

[17] de Bra, P., Houben, G.J. & Wu, H., Aham: A dexter-based reference model for adaptive hypermedia. *ACM Conference on Hypertext and Hypermedia*, Darmstadt, Germany, pp. 147–156, 1999.

[18] Nejdl, W. & Wolpers, M., Kbs hyperbook - a data-driven information system on the web. *Int. WWW8 Conference*, Toronto, 1999.

[19] Henze, N. & Nejdl, W., Adaptivity in the kbs hyperbook system. *Second Workshop on Adaptive Systems and User Modeling on the WWW*, Toronto, 1999.

[20] Rosa Maria Carro, P.R., Estrella Pulido, Tangow: a model for internet-based learning. *International Journal of Continuing Engineering Education and Lifelong Learning*, **11(1–2)**, pp. 25–34, 2001.

[21] Cristea, A. & de Mooij, A., Adaptive course authoring: Mot, my online teacher. *ICT-2003, IEEE LTTF Int. Conference on Telecommunications, Telecommunications + Education Workshop*, 2003.

[22] de Bra, P., Santic, T. & Brusilovsky, P., Aha! meets interbook, and more... *AACE ELearn'2003 Conference*, Phoenix, AZ, pp. 57–64, 2003.

[23] Gilbert, J. & Han, C., Adapting instruction in search of 'a significant diference'. *Journal of Network and Computer Applications*, **22**, 1999.

[24] Triantafillou, E., Pomportsis, A. & Georgiadou, E., Aes-cs: Adaptive educational system based on cognitive styles. *Proceedings of the AH2002 Workshop*, Malaga, Spain, pp. 10–20, 2002.

[25] Stash, N., Cristea, A. & de Bra, P., Authoring of learning styles in adaptive hypermedia: Problems and solutions. *Proc. of the WWW2004 Conference*, New York, NY, pp. 114–123, 2004.

[26] Dunn, R. & Dunn, K., *Teaching students through their individual learning styles: A practical approach*. Reston Publishing, 1978.

[27] Felder, R. & Silverman, L., Learning and teaching styles in engineering education. *Journal of Engineering Education*, **78(7)**, pp. 674–681, 1988.

[28] Kolb, D., *Experiential learning experience as the source of learning and development*. Prentice Hall, 1984.

[29] Honey, P. & Mumford, A., *The manual of Learning Styles*. Peter Honey, Maidenhead, 1992.

[30] Witkin, H., Moore, C., Goodenough, D. & Cox, P., Field-dependent and field-independent cognitive styles and their educational implications. *Review of Educational Research*, **47(1)**, pp. 1–64, 1977.

[31] Aroyo, L. & Dicheva, D., Aims: Learning and teaching support for www-based education. *Int Journal for Continuing Engineering Education and Life-long Learning*, **11(1/2)**, pp. 152–164, 2001.

Index