Adaptive Hypermedia on the Web: Methods, Technology and Applications

(invited talk)

Prof. dr. Paul De Bra*
Department of Computing Science
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: debra@win.tue.nl

Abstract: World Wide Web lets information providers (potentially) reach millions of people located all over the world. These users form a very diverse crowd. No single presentation of the information an organization wishes to distribute will be suitable for all types of users that organization wishes to address. Adaptive hypermedia techniques make it possible to provide each individual user with the information and presentation he or she needs, based on an internal representation of that user, called the *user model*. The user's goals, preferences and needs may change over time. An adaptive Web-site observes these changes, updates the user model on the fly and uses it to adapt the presentation to the changing circumstances.

This paper gives a very brief overview of some existing adaptive hypermedia methods and techniques. Both the (textual or multimedia) content of pages and the links between them can be adapted. The paper indicates which Web technologies are (sometimes or typically) used to realize adaptive Web sites. This includes both technical hacks to make it work and authoring tools intended for subject-matter experts who are not (necessarily) very computer or hypermedia savvy.

Through examples of applications we show reasons for making a Web site adaptive in certain ways. Adaptivity must be used with caution in order to avoid Escher effects that might turn browsing through an adaptive Web site into an adventure game instead of making it easier and more effective.

keywords: adaptive hypermedia, adaptive Web sites, user modeling, adaptive presentation, adaptive navigation, adaptive authoring tools.

1. Introduction

World Wide Web is a great medium for potentially reaching millions of people located all over the world. This potential has been exploited by thousands of companies, research organizations and educational institutes. Web sites have been created for informing customers and business partners, research colleagues, students, visitors and many other categories of users. More and more Web site builders are beginning to realize that one uniform Web site cannot offer the information these millions of different users need, and certainly not offer it in a way that is best suited for each individual user. This is where adaptive hypermedia comes into play. Using an internal representation of a user, called the *user model*, a hypermedia system or Web server can change the information and its presentation according to a user's individual goals, preferences and/or needs.

Adaptive hypermedia (or AH for short) is a research and development field which has its roots in the crossroad of hypermedia and intelligent tutoring systems. As a result, a majority of adaptive hypermedia systems (or AHS), beit Web-based or not, are used for educational applications, and especially for interactive course texts that allow access to more advanced learning material as the student gains knowledge. However, as the Web is

* Paul De Bra is also affiliated with the University of Antwerp and the "Centrum voor Wiskunde en Informatica" in Amsterdam.

being used more and more by companies for promoting and providing information about their products and services, the need arises for providing each individual user with the most appropriate information, in the most desirable form. For instance, browsing through a paper mail-order catalog can be done much faster and more effectively than browsing through an electronic version of that catalog. However, most of the catalog is of no interest to a particular user, and it is time consuming to find the appropriate offerings among the wealth of irrelevant stuff that is also for sale. When an electronic catalog knows what the user's interests are, and possibly what goal the user has during a specific browsing session, it can present the appropriate information very efficiently, thus saving the user some valuable time, and not distracting the user in a way that may lead to no purchase at all. Companies have been formed, and more are coming, that specialize in gathering a lot of information about an individual user, so that this rich user model can be used for adaptive access to Web sites. In this paper we concentrate on techniques that are used for creating adaptive Web sites. We ignore the obvious social and legal privacy issues that arise when user models are exchanged between Web sites (of different companies) or requested by authorities for tracking an individual's Internet activities.

The Web was not designed with adaptive hypermedia in mind. As a result, the standards defined for Web technology, including HTML and HTTP, sometimes hinder the creation of adaptive documents. In the future, dynamic HTML (or DHTML for short) may offer a solution to the need for alternative presentations of content. Since DHTML involves a combination of "pure" HTML, Style Sheets (using CSS) and Javascript or VBscript code we will need sophisticated authoring tools for generating DHTML automatically, as it is impossible for non computer scientists to write DHTML documents manually. HTTP is also lacking some feature(s) that are needed to make adaptive presentation work under all circumstances. We will describe possible ways to program around the Web's shortcomings in order to realize adaptive Web applications.

This paper is structured as follows: Section 2 [Classification] gives a classification of adaptive hypermedia techniques. Section 3 [User Modeling] describes how systems may store information about users in order to do adaptation. It also relates this representation to the issue of authoring. Section 4 [Implementation] shows how adaptivity can be achieved on the Web.

2. Classification of adaptive hypermedia methods and techniques

Adaptive hypermedia (AH) is a term used in different meanings throughout the hypermedia literature. To avoid any confusion we distinguish:

- **Adaptable hypermedia systems** allow the user to explicitly set preferences, or provide a profile through filling out a form. This information is stored in a *user model*. The presentation of the information is then adapted to that model, which is only updated upon explicit request by the user. Some systems may have very elaborate user models, while others only distinguish between a few stereotype users such as "beginner", "intermediate" and "expert",or "student", "employee" and "visitor".
- **Adaptive hypermedia systems (AHS)** build a *user model* by observing the user's navigation through the information space, and in educational systems also by means of (mostly multiple-choice) tests. The presentation is adapted to the user model, and the user model is constantly updated as the user reads the information.

Most AHS are also adaptable, because they need a way to initialize the user model, or to allow users to explicitly adjust that model. However, we shall concentrate on the adaptive features of the systems and applications.

In his overview article Brusilovsky [Brusilovsky 96] describes which AH techniques exist. We briefly recall this classification and refine it as was previously done in [De Bra & Calvi 98]. We also indicate sample applications or application areas where such techniques are (potentially) useful.

Adaptive presentation is the general term for all techniques that adapt the content of a Web page according to the user model. Brusilovsky [Brusilovsky 96] distinguishes:

- **Adaptive text presentation:** the textual content of Web pages is altered depending on the user model. Technical terms can be avoided for beginners, and additional explanations added. Likewise, additional details can be provided for advanced users who are capable of understanding them. Three techniques for creating adaptive text presentation are:
  - *page variants*: several (or a few) different versions exist for a given page. This technique fits a stereotype-based AH system, in which the variation is limited.
  - *fragment variants*: different versions of fragments of a page exist. Fragments can be as small as a single word (like a technical term which can be replaced by a common term which means almost the same), and as large as a whole page. (Thus, page variants are a special case of fragment variants.) The number of variations of a page can be large since each fragment can be adapted to different variables in the user model.
  - *frame based techniques*: using natural language techniques pages are assembled from small information items like words and parts of sentences.
- **Adaptive multimedia presentation:** although this term suggests that multimedia content elements might be adapted to the individual user, current implementations are limited to media selection. Unlike with text, the content of an animation, audio or video fragment is not adapted.

Adaptive navigation is the term for techniques that manipulate the links that are available to the user (at any given moment in time). Brusilovsky [Brusilovsky 96] and De Bra and Calvi [De Bra & Calvi 98] distinguish:

- **Direct guidance:** a link is provided to whichever page the system considers to be most appropriate for this user to go to next. The repeated use of such a "next" button is like following a personalized guided tour.
- **Adaptive sorting of links:** instead of providing a single "best" link this technique provides a list of links, in descending order of relevance. Information retrieval (searching) systems typically provide this kind of list of links. But in educational AHS an adaptively sorted list of links can also be provided to provide guidance as to which pages to read to learn about a certain concept. In [Weber & Specht 97] adaptive "curriculum sequencing" was found to reduce the number of navigation steps a user needs to learn a concept.
- **Adaptive link hiding:** in order to avoid an abundance of links the AHS hides links to currently non-relevant information by making the link anchor (the text which is normally underlined and/or displayed in a different color) indistinguishable from normal text.
- **Adaptive link removal:** this technique removes the link anchor for non-relevant links. This can only be done when the surrounding text still makes sense after removing the link. In a list of items one can easily remove items. In [Brusilovsky & Pesin 95] experiments with ISIS-Tutor are described, where links to non-relevant items were removed. While adaptive link removal is effective in reducing the number of navigation steps, preliminary evaluations [Calvi & De Bra 97] show that users dislike this technique.
- **Adaptive link disabling:** the link functionality is removed but the link text remains. This technique is often combined with link hiding, because if the appearance of an anchor remains, but the link does not "work" the user may think there is a system error. Users dislike adaptive link disabling [Calvi & De Bra 97] but still prefer it over adaptive link removal.
- **Adaptive link annotation:** links are annotated to indicate their relevance. This can be done by changing the presentation (e.g. color) of the anchor itself or by providing another visual cue like a colored dot or arrow. Several systems developed by Brusilovsky, together with others, use adaptive link annotation, combined with other techniques. These systems include ISIS-Tutor [Brusilovsky & Pesin 94], ELM-ART [Brusilovsky et al. 96a] and Interbook [Brusilovsky et al. 96b]. Experiments (see e.g. [Eklund & Brusilovsky 98]) have lead to the conclusion that adaptive link annotation is helpful in reducing the number of navigation steps and in improving comprehension of the learning material.
- **Map adaptation:** a few hypertext systems provide graphical overviews of the link structure of a hyperdocument. This map can be adaptively filtered to present a manageable overview of the parts of the hyperdocument that are relevant to the user.

In [Calvi 98] there is a discussion about the benefits and drawbacks of some of these adaptive linking techniques. Experiments and preliminary evaluations (see e.g. [Eklund & Brusilovsky 98] and [Calvi & De Bra 97]) lead to inconclusive advice: the goal to increase user satisfaction can be served through adaptive link annotation, whereas for improving the efficiency of navigation link removal and/or disabling work better.

3. User Modeling in AH

AHS need an internal representation (or approximation) of the user's state of mind in order to adapt the presentation and navigation to that user. The AHS needs to know the user's goals and task, background, experience in the subject matter, experience in hypermedia navigation, preferences like media, verbosity, etc. Some of this information, namely preferences and sometimes also goals and task AHS from the user's browsing behavior.

Information about the user's state of mind is typically implemented through an *overlay model*. The user's state is compared to a set of variables. Because of the focus of the field of AH on educational systems these variables are often called *concepts*. We shall also use this terminology, but keep in mind that these variables can have any meaning and use, other than representing the user's knowledge about a certain concept.

We distinguish three (sub)types of knowledge representation, differing in the data type used for the concepts:

- In the **Boolean** model there are two possible values for each concepts: *true* and *false*, which could be interpreted as *known* and *not known*. However, the values may actually mean something completely different, like *verbose* and *terse* to indicate whether a user wants a lengthy or short version of each page. Some AH courseware developed by this author uses the Boolean model.
- In the **discrete** model there are a small number of values for each concept. (The Boolean model is a special case of the discrete model.) Systems like ELM-ART [Brusilovsky et al. 96a] and Interbook [Brusilovsky et al. 96b] use a discrete model with four values, meaning *not known*, *learned*, *well learned*, and *known*. A concept becomes *learned* when the user has read about it while she was not ready for it; it becomes *well learned* when the user has read about it while she was ready for it, and it becomes *known* when the user has passed a test about the concept.
- In the **continuous** model there is a range of values, e.g. the real numbers between 0 and 1. Values can be interpreted as *how much* a user knows about a concept, or alternatively as *the chance that* the user knows a concept. The model used by Pilar da Silva [Pilar da Silva 98] comes close to being a continuous model. Instead of values between 0 and 1 it uses percentages (between 0 and 100), thus providing many more possible values per concept than any other discrete model.

It will be clear that the possibilities for adapting a hyperdocument to each individual user are greater in the continuous model than in the discrete model or the Boolean model. However, in order to exploit these possibilities the author (and authoring software) must be able to deal with such a rich value set. In hyperdocuments authored for the Boolean model there is typically a one-to-one relationship between concepts and Web-pages. The user learns about a concept by reading a (single) page. The discrete model may relax this constraint a little bit, for instance by associating a Web page and a test with a single concept. Both techniques lead to a *fine grained* user model, with many detailed concepts. In the continuous model it is possible to associate may pages with a single concept. Each page contributes towards the knowledge of that concept. The concepts may be *coarse grained*, meaning that they represent fairly global knowledge items. This makes it easier to export the knowledge from one AHS to another, assuming that other AHS may not be able to interpret the detailed distribution of knowledge over a large set of (small) concepts.

Authoring AH documents requires that the writer first construct a global map of how he wants the knowledge to be presented to the user. This means determining which concepts will make up the user model, and how these concepts depend on each other or maybe contribute towards knowledge about each other. While it may help the author a bit when the system provides a (graphical) tool for editing this concept map, the main benefit of such a tool is that it helps the system in automating the dependencies. In order to use the concept map each concept must be tied to the pages dealing with that concept. To some extent this could be done automatically through word frequency techniques used in information retrieval, see e.g. [Golovchinsky 97]. However, it will be mostly the author who gets to decide how pages relate to concepts. Determining exactly what the contribution of a single Web-page is to a concept for which there are many pages is difficult.

4. Implementing AH on the Web

In order to provide AH on the Web the server needs to be able to generate Web-pages (in HTML) that are adapted to each individual user. This is typically achieved through a server-side scripting technique such as *servlets* or *CGI-scripts*. In the sequel we call the Web-server plus its scripts the AH engine. The source for the pages, as written by the author, need not necessarily be HTML. The AH engine can translate different source formats to HTML on the fly. This is especially useful for authors who are not used to writing HTML documents. However, the difficulties in creating adaptive hyperdocuments are the same regardless the source language:

- Most Web-pages contain links to other pages. The author must indicate which links are subject to adaptation and which are not. The AH engine manipulates the adaptive links using one of the techniques from [Classification]. In Interbook [Brusilovsky et al. 96b] for instance adaptive links are annotated by means of a colored ball or arrow (which follows the link anchor), while in AHA [De Bra & Calvi 97], [De Bra & Calvi 98] the link anchor text is displayed in blue, purple or black depending on the relevance of the link. (The link is thus annotated or hidden depending on the situation.)
- Some systems provide additional guidance through an annotated table of contents or a textual or graphical representation of (part of) the user model. Such information is often generated automatically, but the author needs to be aware of what information will be displayed. An author who knows this generated information can avoid to repeat it unnecessarily in the information pages.
- The content of the authored Web-pages may need to be adaptive as well. Technical terms should be avoided for users who have not studied their definition. Additional details may be provided for users who have enough experience in the topic at hand. Variations can be achieved by using the *fragment variants technique*, described in [Classification]. This technique requires that the authoring environment provides a facility for indicating which variation of each fragment is displayed, depending on which condition on the variables in the user model.

Currently, tools for authoring adaptive text presentations in an easy way are still lacking. AHA [De Bra & Calvi 97], [De Bra & Calvi 98] for instance uses "*if statements*" encoded as HTML-comments. This technique is convenient for authors who are familiar with authoring HTML and with logic. The majority of authors of Web-pages will most likely not be able to use such an authoring environment.

Implementing an AH engine is also non-trivial. In an AHS it is common that when the user revisits a page the appearance of that page may (need to be) different. Links may be sorted or annotated differently, text fragments may have changed, a "next" button may point to a different location, etc. While the changes are usually subtle, in order to avoid Escher effects, the architecture of the Web makes it difficult to offer even small variations to pages: the HTTP standard (version 1.1) states:

> *By default, the Expires field does not apply to history mechanisms. If the entity is still in storage, a history mechanism should display it even if the entity has expired, unless the user has specifically configured the agent to refresh expired history documents.*
> (quoted from RFC's 1945 and 2068, which define HTTP/1.0 resp. 1.1)

Unfortunately, the "agent" (meaning the Web-browser) is not required to offer the option of refreshing expired documents when going back to them. Luckily the two most popular browsers (Netscape Navigator and Microsoft Internet Explorer) do request expired documents from the server each time they are revisited. (However, due to a bug in Netscape Navigator, it may be necessary to disable the browser's memory and disk cache when going through a proxy gateway. This adversely effects the overall server load and browser performance.) Older browsers, including NCSA Mosaic for X (the browser that made the Web popular, long before Netscape Navigator and Internet Explorer existed) may refuse to revisit pages that are available through the browser's history mechanism.

When a Web-based AHS uses HTML frames (a presentation technique introduced by Netscape and later incorporated in the HTML 4.0 standard), following a link within one frame may result in the other frames no longer corresponding to the actual user model. In order to update the other frames, a small piece of Javascript or VBscript code must be added to each page, requesting the other frames to be updated if needed. The two most popular browsers support Javascript, but some others may not.

Adaptive link annotation, and link hiding, can be done by changing the color of the link text. This requires the use of HTML style sheets, a technique introduced with the HTML 3.2 standard. Again, not yet all browsers support style sheets.

5. Conclusions and final remarks

There exist many different techniques for adapting the presentation of information and the navigation between pages on the Web. Several applications have been evaluated and show the benefits of the different adaptive hypermedia techniques. However, there is no evidence that the findings of these experiments can be generalized to all applications that use these techniques. In fact, the process of constructing a user model, determining dependencies between concepts, and tying concepts to Web-pages is not yet automated. While adaptive hyperdocuments *can* be created that are easier to use than static equivalents, the techniques can equally well be used to make navigation through a hyperdocument more difficult. There currently exist no (software) tools for verifying the usability of adaptive hyperdocuments. Therefore, user evaluation remains the most important way to determine the effectiveness of adaptivity for a particular application.

6. References

[Brusilovsky 96] Brusilovsky, P. (1996). Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, **6**: 87-129, Kluwer academic publishers.

[Brusilovsky & Pesin 94] Brusilovsky, P., Pesin, L. (1994). ISIS-Tutor: An intelligent learning environment for CDS/ISIS users. *Interdisciplinary workshop on complex learning in computer environments*, 29-33.

[Brusilovsky & Pesin 95] Brusilovsky, P., Pesin, L. (1995). Visual annotation of links in adaptive hypermedia. *CHI'95 Conference Companion*, ACM Press, 222-223.

[Brusilovsky et al. '96a] Brusilovsky, P., Schwarz, E., Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. *Third International Conference on Intelligent Tutoring Systems, ITS-96*. (Lecture Notes in Computing Science, **1086**, 261-269).

[Brusilovsky et al. '96b] Brusilovsky, P., Schwarz, E., Weber, G. (1996). A Tool for Developing Adaptive Electronic Textbooks on WWW. *AACE WebNet'96 Conference*, 64-69.

[C98] Calvi, L. (1998). A Proficiency-Adapted Framework for Browsing and Information Filtering in Web-Based Educational Systems. Methodological Implications for Language Learning on the WWW. Doctoral Thesis, University of Antwerp.

[Calvi and De Bra 97] Calvi, L., De Bra, P. (1997). Proficiency-Adapted Information Browsing and Filtering in Hypermedia Educational Systems, *User Modeling and User-Adapted Interaction*, **7**: 257-277, Kluwer academic publishers.

[De Bra and Calvi 97] De Bra, P., Calvi, L. (1997). Creating adaptive hyperdocuments for and on the Web. *AACE WebNet'97 Conference*, 149-165.

[De Bra & Calvi 98] De Bra, P., Calvi, L. (1998). AHA: A Generic Adaptive Hypermedia System. *2nd Workshop on Adaptive Hypertext and Hypermedia*, 5-11.

[Eklund and Brusilovsky 98] Eklund, J., Brusilovsky, P. (1998). The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence. *2nd Workshop on Adaptive Hypertext and Hypermedia*, 11-17.

[Golovchinsky 97] Golovchinsky, G. (1997). What the Query Told the Link: The integration of hypertext and information retrieval. *ACM Hypertext conference*, 67-74.

[Pilar da Silva '98] Pilar da Silva, D. (1998). Concepts and documents for adaptive educational hypermedia: a model and a prototype. *2nd Workshop on Adaptive Hypertext and Hypermedia*, 33-43.

[Weber and Specht 97] Weber, G., Specht, M. (1997). User modeling and adaptive navigation support in WWW-based tutoring systems. *6th International Conference on User Modeling*, 289-300.