In: C. Nicholas and J. Mayfield (eds.): *Intelligent hypertext: Advanced techniques for the World Wide Web*. Lecture Notes in Computer Science, 1326, Berlin: Springer-Verlag, 12-30.

Efficient Techniques for Adaptive Hypermedia

Peter Brusilovsky

Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
E-mail: plb+@andrew.cmu.edu

Abstract. Adaptive hypermedia is a new direction of research within the area of adaptive and user model-based interfaces. Adaptive hypermedia (AH) systems build a model of the individual user and apply it for adaptation to that user, for example, to adapt the content of a hypermedia page to the user's knowledge and goals, or to suggest the most relevant links to follow. AH systems are now used in several application areas where the hyperspace is reasonably large and where a hypermedia application is expected to be used by users with different goals, knowledge and backgrounds. This chapter provides a brief survey of existing adaptive hypermedia techniques. Special attention is paid to the techniques implemented in the World Wide Web and to techniques which have been approved by an experimental study and shown to be effective. Among few others approved techniques we describe adaptive annotation techniques developed by our group at the Moscow State University.

1 Introduction

The use of adaptive hypermedia (AH) systems is one way to increase the functionality of hypermedia. AH systems can be useful in any situation when the system is expected to be used by people with different goals and knowledge and where the hyperspace is reasonably big. Users with different goals, knowledge, and backgrounds may be interested in different pieces of information presented on a hypermedia page and may use different links for navigation. AH tries to use knowledge about a particular user, represented in the user model, to adapt the information and links being presented to that user. Adaptation can also protect the user from being lost in hyperspace. Knowing user goals and knowledge, AH systems can support users in their navigation by limiting the browsing space, suggesting the most relevant links to follow, or providing adaptive comments to visible links.

The goal of this chapter is to present the ideas of adaptive hypermedia to a general hypermedia and WWW-oriented audience and to show some ways to implement AH systems. The main content of the chapter is a selective survey of adaptive hypermedia techniques (a complete review of AH techniques can be found in (Brusilovsky, 1996)). In selecting the techniques to present, we gave first priority to "efficient" techniques (i.e., to the techniques which have been validated by experiment), second priority to the techniques which were implemented in World Wide Web (WWW) context, and third priority to the techniques similar to the techniques of the two above groups. The survey itself is presented in the Section 3. Before presenting the survey, in Section 2, we discuss the issue "what can be adapted in adaptive hypermedia". From this point of view, we provide a simple classification of existing adaptive hypermedia techniques distinguishing two major ways to adapt hypermedia (adaptive presentation and adaptive navigation support) and five adaptive hypermedia technologies. After the survey, in Section 4, we review briefly the most important reported experiments with adaptive hypermedia. These experiments provide guidelines for the designers of new AH systems showing what adaptive hypermedia *can* and what *cannot* do. A reasonable part of Sections 3 and 4 is devoted to several adaptive hypermedia techniques designed by our group in the Moscow State University. In conclusion we summarize main arguments of the chapter.

2 What Can be Adapted in Adaptive Hypermedia

At some level of generalization, hypermedia consist of a set of nodes or "pages" connected by links. Each page contains some local information and a number of links to related pages. These links can appear within the content of a page, in a separate menu, or on a separate local map. Hypermedia systems can also include an index and a global map which provide links to all accessible pages. What can be adapted in adaptive hypermedia are the content of regular pages (content-level adaptation) and the links from regular pages, index pages, and maps (link-level adaptation). As a rule, content-level adaptation is used to solve the problem of hypermedia systems which are used by different classes of users, while link-level adaptation is used to provide navigation support and prevent users from getting lost in hyperspace. We distinguish content-level and link-level adaptation as two different ways of hypermedia adaptation and call the former *adaptive presentation* and the latter *adaptive navigation support*.

2.1 Adaptive Presentation

The idea of various adaptive presentation techniques is to adapt the content of a page accessed by a particular user to current knowledge, goals, and other characteristics of the user. For example, a qualified user can be provided with more detailed and deep information while a novice can receive additional explanations. Existing adaptive presentation techniques deals with text adaptation. Text adaptatioin implies that different users in different time may get different texts as a content of the same page. We group all these techniques into one technology which we call adaptive text presentation technology. Currently, this technology is the most studied technology of hypermedia adaptation. Most part of the early works on adaptive hypermedia was centered around adaptive text presentation (Beaumont, 1994; Boyle and Encarnacion, 1994; Brusilovsky, 1992; de Rosis et al., 1993; Fischer et al., 1990). This direction of research was influenced by the research on adaptive explanation and adaptive presentation in intelligent systems (Moore and Swartout, 1989; Paris, 1988). As we will show in the following sections, a number of different techniques for adaptive text presentation have been developed.

2.2 Adaptive Navigation Support

The idea of adaptive navigation support techniques is to help users to find their paths in hyperspace by adapting link presentation to the goals, knowledge, and other characteristics of an individual user. Though this area of research is new, a number of interesting techniques have been already suggested and implemented. These techniques can be classified in several groups according to the way they adapt presentation of links. We consider these groups of techniques as different technologies for adapting link presentation. The most popular technologies are *direct guidance*, *sorting*, *hiding*, and *annotation.*

Direct guidance is the most simple technology of adaptive navigation support. Direct guidance can be applied in any system which can decide what is the next "best" node for the user to visit according user's goal and other parameters represented in the user model. To provide direct guidance, the system can outline visually the link to the "best" node as it is done in Web Watcher (Armstrong et al., 1995), or present an additional dynamic link (usually called "next") which is connected to the "best" node as in ISIS-Tutor (Brusilovsky and Pesin, 1994), SHIVA (Zeiliger, 1993), HyperTutor (Pérez et al., 1995), and Land Use Tutor (Kushniruk and Wang, 1994). The former way is clearer; the latter is more flexible, because it can be used to recommend the node that is not connected directly to the current one (and not represented on the current page). A problem of direct guidance is that it provides no support for the users who would not like to follow the system's suggestions. Direct guidance is useful but it should be used together with a "more supportive" technology.

The idea of *adaptive ordering* technology is to sort 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 ordering has a limited applicability: it can be used with non-contextual links, but it can hardly be used for indexes and content pages (which usually have a stable order of links), and can never be used with contextual links and maps. Another problem with adaptive ordering is that this technology makes the order of links non-stable: it may change each time the user enters the page. Recent research shows that the stable order of options in menus is important for novices (Debevc et al., 1994; Kaptelinin, 1993). However, this technology shows to be useful for information retrieval (IR) applications (Armstrong et al., 1995; Kaplan et al., 1993; Mathé and Chen, 1996). As we will see in Section 4, experimental research (Kaplan et al., 1993) showed that adaptive ordering can significantly reduce navigation time in IR applications where each page can have many non-contextual links. A related application area where adaptive ordering can be used is on-line documentation systems (Hohl et al., 1996).

Hiding is currently the most commonly used technology for adaptive navigation support. The idea of navigation support by hiding is to restrict the navigation space by hiding links to irrelevant pages. A page can be considered as not relevant for several reasons: for example, if it is not related to the user's current goal (Brusilovsky and Pesin, 1994; Höök et al., 1996; Vassileva, 1996) or if it presents materials which the user is not yet prepared to understand (Brusilovsky and Pesin, 1994; Gonschorek and Herzog, 1995; Pérez et al., 1995). Hiding protects users from the complexity of the unrestricted hyperspace and reduce their cognitive overload. Hiding has a wide applicability: it can be used with all kinds of non-contextual, index, and map links by hiding buttons or menu items (Brusilovsky and Pesin, 1994), and with contextual links by replacing clickable "hot words" to normal text (Gonschorek and Herzog, 1995; Pérez et al., 1995). Hiding is also more transparent to the user and looks more "stable" than adaptive ordering (links are usually added incrementally, but not removed or reordered). Hiding has, however, another problem: as noted by some psychologists, hiding can provoke formation of incorrect mental models of the hyperspace.

The idea of *adaptive annotation* technology is to augment the links with some form of comments which can tell the user more about the current state of the nodes behind the annotated links. These annotations can be provided in textual form (Zhao et al., 1993) or in the form of visual cues using, for example, different icons (Brusilovsky et al., 1996a; de La Passardiere and Dufresne, 1992), colors (Brusilovsky and Pesin, 1994; Brusilovsky and Zyryanov, 1993), font sizes (Hohl et al., 1996), or font types (Brusilovsky et al., 1996a). Link annotation is known to be an effective way of navigation support in hypermedia (Zhao et al., 1993). The typical kind of annotation considered in traditional hypermedia is static (user independent) annotation. Adaptive navigation support can be provided by dynamic user model-driven annotation. Adaptive annotation in its simplest history-based form (outlining the links to previously visited nodes) has been applied in some hypermedia systems including several World-Wide Web browsers. Even this simplest form of adaptive annotation which can distinguish only two states of links (links to visited/not visited nodes) appears to be quite useful. Current adaptive hypermedia systems (Brusilovsky and Pesin, 1994; Brusilovsky et al., 1996a) can distinguish and annotate differently up to six states on the basis of the user model.

Annotation seems to be a good way of adaptive navigation support. Annotation can be naturally used with all possible forms of links. This technique supports stable order of links and avoids problems with incorrect mental maps. Annotation is generally a more powerful technology than hiding: hiding can distinguish only two states for the related nodes - relevant and non relevant - while annotation, as mentioned above, can distinguish up to six states (for example, Hypadapter (Hohl et al., 1996) uses annotations to show several levels of relevancy). Annotations do not restrict cognitive overload as much as hiding does, but hiding technology can be quite well simulated by the annotation technology using a kind of "dimming" instead of hiding for "not relevant" links. Dimming can decrease cognitive overload in some extent (the user can learn to ignore dimmed links). Dimmed links are still visible though (and traversable, if required) which prevents the user from forming wrong mental maps.

Direct guidance, sorting, hiding, and annotation are the primary technologies for adaptive navigation support. As we will see in the following sections, most existing adaptation techniques use exactly one of these ways to provide adaptive navigation support. However, these technologies are not mutually exclusive and can be used in combinations. For example, ISIS-Tutor (Brusilovsky and Pesin, 1994) uses direct guidance, hiding, and annotation; Hypadapter (Hohl et al., 1996) uses sorting, hiding, and annotation. In particular, the direct guidance technology can be naturally used in combination with any of the other three technologies.

3 Adaptive Hypermedia Techniques

3.1 Adaptive Presentation Techniques

A simple, but effective technique for content adaptation is the "conditional text" technique which is used in ITEM/IP (Brusilovsky, 1992), Lisp-Critic (Fischer et al., 1990), and C-book (Kay and Kummerfeld, 1994). With this technique, all information about a concept is divided into several chunks of texts. Each chunk is associated with a condition on the level of user knowledge represented in the user model. When presenting the information about the concept the system presents only the chunks where the condition is true. This technique is a low-level technique (it requires some "programming" work from the author to set all the required conditions) but it is also very flexible. By choosing appropriate conditions on the knowledge level of the current concept and related concepts represented in the user model the author can implement different methods of adaptation. A simple example is hiding chunks that contain irrelevant explanations if the user's knowledge of the current concept is good enough, or turning on a chunk with comparative explanations if the corresponding related concept is already known. This conditional text technique can be easily applied in the WWW context. The work of Kay and Kummerfeld (1994) presents several ideas which can be used to implement this technique in a WWW-based system.

A higher level technique which can also turn off and on different parts of the content according to the user knowledge level is suggested in MetaDoc (Boyle and Encarnacion, 1994) and developed further in KN-AHS (Kobsa et al., 1994). This technique is based on *stretchtext* which is a special kind of hypertext. In a regular hypertext, a result of clicking on a hot word is moving to another page with related text. In stretchtext this related text can simply replace the activated hot word (or a phrase including the hot word), thereby extending the text of the current page. If required, this extended or "uncollapsed" text may be collapsed back to a hot word. Each node in MetaDoc is a stretchtext page which may contain many "uncollapsable" hot words. The idea of adaptive stretchtext presentation in MetaDoc is to present a requested page with all stretchtext extensions that are non-relevant to the user being collapsed, and all extensions relevant to the user being uncollapsed. To achieve this result an author can declare some uncollapsable textual information contained in a node as an additional explanation of a particular concept, or as a low level detail of a particular concept. Optionally, the user of MetaDoc with good knowledge of a concept will always find additional explanations of this concept hidden (collapsed) and all low level details uncollapsed. On the contrary, the user with poor knowledge of a concept will always find additional explanations of this concept visible and all low level details collapsed. The user with medium level knowledge will see both kinds of information. An important feature of the adaptive stretchtext technique is that it lets both the user and the system adapt the content of a particular page, taking into account both the knowledge and the preferences of the user. After its initial presentation, the stretchtext page can be further adapted by the user who can uncollapse and collapse appropriate explanations and details according to his or her preferences. The system updates the user model according to the preferences demonstrated by the user to ensure that the user will always see the preferred combination of collapsed and uncollapsed parts. For example, if the user has collapsed additional explanations of a particular concept, the system will always show additional explanations of this concept collapsed until the user changes the preferences.

The most powerful of the all content adaptation techniques is the frame-based technique implemented in Hypadapter (Hohl et al., 1996), and EPIAIM (de Rosis et al., 1993). With this technique all the information about a particular concept is represented in the form of a frame. Slots of a frame can contain several different explanations of the concept, links to other frames, examples, etc. Special presentation rules are used to decide which slots should be presented to a particular user and in which order. More exactly, in EPIAIM these rules are used to select one of existing presentation schemes (each scheme is an ordered subset of slots) and the scheme is used to present the concept. In Hypadapter, the rules are used to calculate the "presentation priority" for each slot. Then a subset of slots with high priority are presented in order of decreasing priority. In their conditional parts, these rules can refer not only to the user knowledge of a concept being presented, but also to any feature represented in the user model. In particular, both systems which use this technique take into account the background of the user.

A good example of implementation of the frame-based technique on WWW is PEBA-II system[1] (Milosavljevic et al., 1996). This system uses frame-based knowledge representation and natural language generation techniques to generate descriptive and comparative information about different animals. This information is adapted to the user level of expertize (current version of PEBA-II is rather adaptable then adaptive: users have to select their level of expertize themselves).

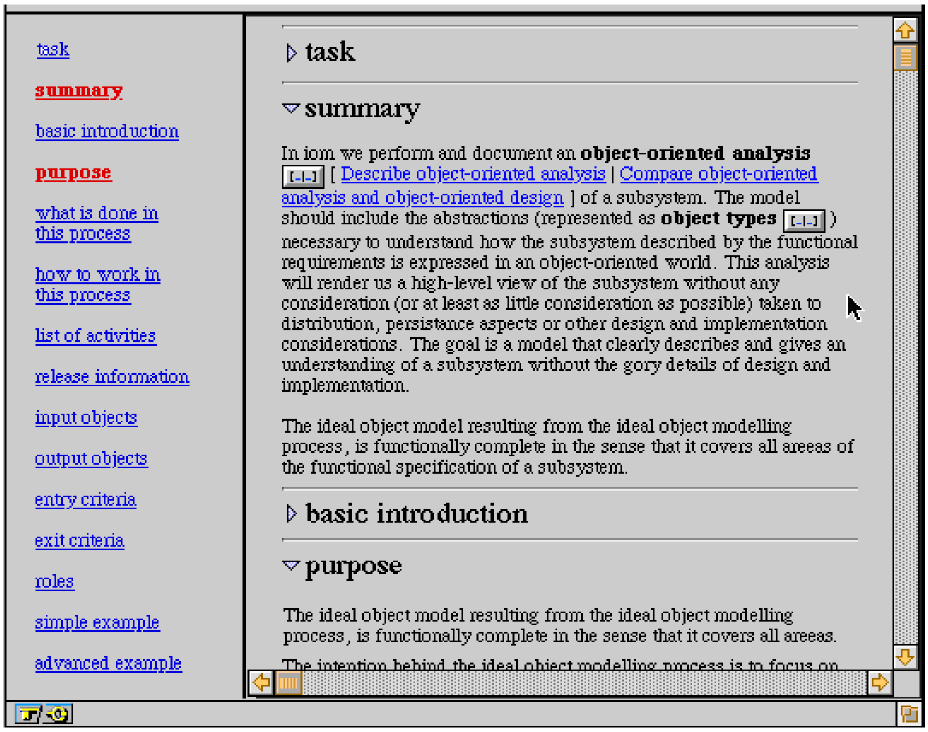

Fig. 1. Adaptive presentation with WWW interface in PUSH

A technique developed recently in the PUSH project (Höök et al., 1996) can be considered to be a combination of stretchtext and frame-based adaptation. A hypermedia page in this on-line information system provides a complete description of a particular object structured as an ordered sequence of typed information entities. Each type of object in PUSH has its own assortments of information entity types which are used to describe an object of that type. This technique is similar to the frame-based model (objects may be considered as frames and information entities as slots). The difference is that each information entity in PUSH is a reasonably large

[1]http://www.mri.mq.edu.au/peba/

portion of hypertext. The complete description of an object is usually itself large, and takes several pages of information. To protect users from information overflow and to help them to find a required piece of information, the system uses hiding: it presents only those types of information about the current object which are relevant to the current goal of the user (the goal can be set by the user or deduced by the system). At the same time, to keep the adaptation transparent, the system maintains the stable presentation order of the information and never hides non-relevant entities completely: the titles of hidden non-relevant entities are always shown. If the user is not satisfied with the system's decision to show or to hide a particular entity, he or she can collapse or uncollapse the content of the information entity by clicking on an icon near its title. The resulting interface looks similar to MetaDoc stretchtext interface: non-relevant pieces of material are not presented to the user, showing only a keyword (in MetaDoc) or the title (in PUSH), but the user can override the adaptation by opening and closing any desired piece of information.

Figure 1 shows an example from the PUSH system of implementing stretchtext-like adaptive presentation technique in WWW context. Generally speaking, a stretchtext-based technique is not the easiest one to implement on WWW. However, present developments in WWW area, such as Java[2] technology open the way to stretchtext-based adaptive presentation techniques on WWW (Espinoza and Höök, 1996)

3.2 Adaptive Navigation Support Techniques

One of the most frequently cited techniques for adaptive sorting of links was implemented in HYPERFLEX (Kaplan et al., 1993), which can be considered to be an on-line information system or IR hypermedia system. HYPERFLEX provides the user with navigation support by displaying an ordered list of nodes related to the current node. The links are ordered according to their relevance to the current node (the most relevant appearing first). In addition, HYPERFLEX maintains a list of possible search goals. If the user selects one of these search goals, sorting takes into account the relevance of the displayed links to the selected goal. New goals can be created by users themselves. The main component of the user model in HYPERFLEX is a relevance matrix which stores relevance values between each pair of documents and from each goal to each document. This matrix reflects mainly user preferences for the link order: in HYPERFLEX a user can move the links to tell the system directly his or her preferences on the relative order of links (i.e., which relevant links and in which order he or she would like to see when viewing a particular document or when pursing a particular goal). These preferences are processed by the system to update the user model. Therefore the preferences selected in one context can influence adaptation in another context.

[2] http://java.sun.com/

An example of an adaptive navigation support technique based on hiding is the rule-based technique implemented in educational hypermedia systems HyperTutor (Pérez et al., 1995) and SYPROS (Gonschorek and Herzog, 1995). Both systems use special sets of pedagogical rules to decide which concepts and nodes should be visible at the given moment and which should not. These rules take into account the type of the concept and its links to other concepts and the current state of user knowledge reflected in the user model. If the system decides not to make a node visible, then all contextual links to this node are changed from hot words to normal text. The rule-based hiding technique is very flexible. By manipulating the text with hiding rules it is possible to implement several navigation support methods based on hiding, such as hiding nodes the user is not yet ready to learn, or hiding nodes which belong to future lessons.

A very different hiding technique is suggested in Hynecosum (Vassileva, 1996). This system supports hierarchies of tasks for users of different categories. Each hyperspace node in the system is indexed by the elementary (terminal) tasks which require access to this node. Thus, for each terminal task the system can compile the associated list of nodes relevant to this task. By definition, a list of relevant nodes for more high level tasks includes all the nodes relevant at least to one of its subtasks. Using this knowledge Hynecosum can provide local orientation support by hiding. In Hynecosum the user directly tells the system which task he or she is currently performing by selecting one of the tasks from a personal hierarchy. After that, the system shows the user only the nodes which are relevant to the current task. It makes the set of visible nodes manageable.

An example of annotation-based adaptive navigation support technique can be provided by our systems ISIS-Tutor (Brusilovsky and Pesin, 1994) and ITEM/PG (Brusilovsky and Zyryanov, 1993). The backbone of the hyperspace in these systems is formed by a conceptual network which represents the pedagogical structure of the subject being taught. Each concept of the conceptual network is represented by a node of the hyperspace. Concept nodes of the network are connected by different kinds of relationships such as "is-a", "part-of", and "prerequisite". Another kind of nodes in the hyperspace are learning units (such as presentation units, problems, and examples). Each learning unit is connected to all concept nodes which are required to work with this unit. The weighted overlay student model represents separately the level of the student's knowledge of each of the concepts. Using this student model and prerequisite links, the systems can distinguish four educational states for each node represented by a hypermedia page: not-ready-to-be-learned (i.e., has unlearned prerequisites), ready-to-be-learned, in-work (learning started), and learned (student has solved the required number of problems for the concept). Our idea is that concepts with different educational states have different meanings for the student and making educational states visible will help the student in hyperspace navigation. To make educational states visible, links to the concepts with different educational states are annotated differently using different colors and some special characters (more details about annotations used in the current version of ISIS-Tutor are given in Section 4.3).

The same framework was used in ISIS-Tutor to implement an adaptive navigation support technique based on the learning goal. In this system, the teacher can set for each student a sequence of learning goals. A goal is a set of concept nodes of the network which must be learned. Concepts which belong to the same goal are expected to be learned together and then mastered by solving a number of problems before a student moves to the next goal. ISIS-Tutor uses two methods to adapt to the learning goal: first, to attract the student's attention, it can outline links to the concepts belonging to the current goal; second (as an option), to decrease the student's cognitive load, it can hide concepts which belong to future learning goals.

Annotation is a very suitable technique for implementing adaptive navigation support on WWW. An example of adaptive annotation on WWW is provided by a tutoring system ELM-ART[3] (Brusilovsky et al., 1996a) and an authoring system InterBook[4] (Brusilovsky et al., 1996b). These systems use the same technique as ISIS-Tutor for knowledge-based and learning goal-based annotation of links. Different are the visual cues which are used by ELM-ART and InterBook: rather then changing the color of the hot words themselves (as was done in ISIS-Tutor and ITEM/PG), these systems change the color of icons and the typeface of the font (for users without color monitors). In particular, for the links to unvisited nodes ELM-ART uses the traffic light metaphor: links to not-ready-to-be-learned nodes are annotated by a red ball icon and italic font, links to ready-to-be-learned nodes belonging to the current goal are annotated by a green ball icon and boldface font, and links to ready-to-be-learned nodes not belonging to the current goal are annotated by a yellow ball icon and normal font (Figure 2). ELM-ART and InterBook implement adaptive annotation of links using Common Gateway Interface[5], Common Lisp Hypermedia Server CL-HTTP[6] (Mallery, 1994) and on-the-fly page generation.

Currently, all major adaptive navigation support technologies are implemented on WWW. ELM-ART and InterBook (Brusilovsky et al., 1996a; Brusilovsky et al., 1996b) provide an example of annotation-based and sorting-based adaptive navigation support. WebWatcher (Armstrong et al., 1995) implements direct guidance technology in information-retrieval context (WebWatcher is not completely adaptive yet: the adaptation is currently based on a group user model rather then on individual user model). Some ideas how to implement hiding-based adaptive navigation support in WWW context can be found in (Lai et al., 1995). The system described there has all the required knowledge to implement hiding of links to not-ready-to-be-learned pages, but rather then hide such links, the system modifies their

---

[3]http://www.psychologie.uni-trier.de:8000/elmart
[4]http://www.contrib.andrew.cmu.edu/~plb/InterBook.html
[5]http://hoohoo.ncsa.uiuc.edu/cgi/
[6]http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html

behavior. If a page behind a link is not-ready-to-be-learned (i.e., it has prerequisite pages not visited by the user), the link drives the user not to the requested page, but to the list of pages which must be learned before viewing the requested page.

Fig. 2. Example of adaptive annotation of links using WWW interface in ELM-ART. The metaphor is traffic lights. Red (italic typeface) means not ready to be learned, green (bold) means ready and recommended (belongs to the learning goal), yellow means ready but not recommended.

4 Evaluation of Adaptive Hypermedia Techniques

The luck of experimental investigation of adaptive hypermedia is currently a weak point of this research direction. Very few reported AH techniques have been validated by a special study (Boyle and Encarnacion, 1994; Brusilovsky and Pesin, 1995; Clibbon, 1995; Höök, in press; Kaplan et al., 1993). In this section we review briefly the most important reported studies.

4.1 Evaluation of an Adaptive Presentation Technique

The most comprehensive evaluation of adaptive presentation in hypermedia was performed by Boyle and Encarnacion (1994) with their system MetaDoc. The goal of the experiment was to compare three kinds of hypertext: normal hypertext, stretchtext (i.e., hypertext extended with stretchtext functionality), and adaptive stretchtext in the context of on-line information access. Two kinds of tasks were used to compare these kinds of hypertext: reading comprehension tasks and navigation tasks. The systems compared were the original MetaDoc with all functionality and two "disabled" versions of MetaDoc: the stretchtext version which had all stretchtext functionality, but no user modeling and adaptation and the hypertext-only version which had no stretchtext functionality at all.

The subjects (computer science students) were randomly assigned to one of the three systems forming three groups: the hypertext group, the stretchtext group and the MetaDoc group. The subjects had some time to learn their systems and to browse the actual document. Then each subject received a booklet with five search and navigation questions and eight reading comprehension questions. The subject was allowed three minutes to find the answer to the search and navigation questions and then five minutes for the reading comprehension questions. For each question the subject was allowed three tries in finding the correct answer. For the search and navigation questions, the subject simply pointed out the location of the answer. For reading comprehension questions, the answer was provided orally.

The main results of the experiment are shown in the Table 1. Analysis of Variance (ANOVA) was the primary statistical test used. For all shown parameters the effect was significant at the 1 percent level. On a paired test a significant difference for reading comprehension time was found between stretchtext and MetaDoc groups. For the reading comprehension correctness and the search time a significant difference was found between hypertext and both other groups, though no significant difference was found between stretchtext and MetaDoc. For the three other parameters related with navigation: search correctness, number of visited nodes (including repetitions), and number of operations, no significant difference was found.

Table 1. Significant results of MetaDoc evaluation

	Hypertext		Stretchtext		MetaDoc	
	Expert	Novice	Expert	Novice	Expert	Novice
Reading comp. time (sec.)	1780	1930	1250	1780	810	1420
Reading comp. correct	5	3	6.5	7	7	7
Mean search time (sec.)	755	725	645	530	555	575

Thus, the experiment has shown that stretchtext-based content adaptation is an efficient adaptation technique which can increase user performance by improving reading comprehension. With this technique, reading comprehension time decreases significantly, without loss in understanding. In fact, understanding even increases, but this effect is possibly provided by the stretchtext technology itself rather then by the adaptation technique. At the same time, content adaptation does not affect user navigation. For all navigation-related parameters including time and number of visited nodes there was no significant difference between adaptive and non-adaptive versions of MetaDoc.

4.2 Evaluation of an Adaptive Navigation Support Technique: Sorting

The first evaluation of adaptive navigation support by sorting was performed by Kaplan et al. (1993) with their system HYPERFLEX. They performed two pilot studies. In the first small study (with four subjects) they examined the usefulness of goal-directed search in the hypertext. The subjects were asked questions relating to information stored in the hypertext. Each user answered ten questions. For five of these questions there existed relevant goals among the system supported goals. That is, the user could select this goal as the current goal and use the adaptively sorted list of links to related nodes as a navigation support. For five other questions no relevant goals were provided. In the version of HYPERFLEX used in this experiment the users were not able to create their own goals. The results of the experiment shown in Table 2, demonstrate that goal-based adaptive sorting seriously decrease search time and the number of searched topics, while the correctness of answers even increased slightly.

Table 2. First pilot study with HYPERFLEX

	Search time	# Topics	% correct
With relevant goal	462 sec.	8.8	83%
No relevant goal	716 sec	12.2	75%

The goal of the second pilot study was to compare the efficiency of two main methods of adaptation in HYPERFLEX: current-node-based adaptation (the user selects the current node of interest and the system orders the relevant links according their relevancy to the current node) and "current goal" based adaptation (the user selects the current goal and the system orders the relevant links according their relevancy to the current goal). Three versions of HYPERFLEX were used in experiment: the version with node-based adaptation only, the version with goal-based adaptation only, and a fully functional system with both kinds of adaptation available. Eighteen subjects participated in the study (six subjects for each version). Each subject was asked to perform four search tasks. Dependent measures were the time to complete each task and the number of topics (nodes) searched for each task. Since all the subjects completed the task successfully, "% correct" was not a relevant measure.

The results shown in Table 3 demonstrate that both methods of adaptation are efficacious, because the users of the fully functional system showed better performance then either of the two other groups.

Table 3. Second pilot study with HYPERFLEX

	Search time	# Topics	Time / topic
With current goal	387 sec.	8.6	45 sec.
With current node	356 sec.	6.8	52 sec.
Fully functional	345 sec	9.0	38 sec.

While the results of both studies should be interpreted with caution due to the small sample size (the original paper contains no data about significance of the results), they show that sorting-based adaptive navigation support can improve user performance in information search tasks.

4.3 Evaluation of Adaptive Navigation Support Techniques: Hiding and Annotation

The first evaluation of adaptive navigation support by hiding and annotation was performed by our group in the Moscow State University. The goal of our study was to check the efficiency of these two adaptation technologies and, in particular, to compare these technologies in an educational context. As a base system, we used for the experiment our ISIS-Tutor system (Brusilovsky and Pesin, 1994) which was briefly described above.

The current version of ISIS-Tutor uses adaptive annotation as a primary technique for adaptive navigation support. As an optional mode of work, ISIS-Tutor also implements adaptive hiding of links. The idea of hiding in ISIS-Tutor is to reduce the cognitive load by hiding from students all links to nodes which they are "not expected to learn". There are two kinds of hidden nodes in ISIS-Tutor: not-ready-to-be-learned nodes and ready-to-be-learned nodes that are outside the current educational goal. In normal annotation mode, the links to these nodes are not specially annotated. In hiding mode, these links are hidden. Note that hiding mode in ISIS-Tutor is more advanced then typical hiding. It is a combination of hiding and annotation, because learned, in-work and ready-to-be-learned nodes are still annotated as in normal annotation mode.

In our experiment, we used three versions of ISIS-Tutor: a non-adaptive version "A" which provided neither annotation nor hiding; a normal version "B" with adaptive annotation; and a version "C" that worked in hiding mode. In adaptive versions of ISIS-Tutor links to not-ready-to-be-learned nodes were not specially colored, ready-to-be-learned were colored red, both in-work and learned were colored green, and learned concepts was additionally marked with a "+" sign (we used special signs to avoid using too many colors). Links to nodes which are within the current educational goal were marked with a "-" sign. Links to not-ready-to-be-learned nodes and nodes outside the current educational goal were not specially annotated in version B and hidden in version C. The summary of different annotations applied in these three versions is shown in the table 4.

Table 4. Summary of annotations applied in three versions of ISIS-Tutor.
NA means no annotation.

	A. Non-adaptive	B. Adaptive annotation	C. Adaptive hiding
Outside educational goal	NA	NA	hidden
Within educational goal	NA	mark "-"	mark "-"
Well-learned	NA	mark "+"	mark "+"
Known	NA	green color	green color
Ready-to-be-learned	NA	red color	red color
Not-ready-to-be-learned	NA	NA	hidden

Twenty-six subjects (first year computer science students of the Moscow State University) took part in the experiment. They were briefly introduced to ISIS-Tutor and then had up to 45 minutes to work with the system. The same educational goal (ten concepts and ten test problems) was given to each student. To complete the course, each user had to solve all ten problems. The subjects were divided randomly into three groups. Group A worked with version A (non-adaptive version). Group B worked with version B (adaptive annotation). Group C worked with version C (adaptive annotation and hiding). All actions of students working with the system were recorded and then analyzed to compare various aspects of user performance. The most important dependent variables were the time required to complete the course and the overall number of navigation steps. According to the results of the experiment with HYPERFLEX, we expected that both the time and the number of steps would decrease for adaptive versions.

Table 5. Results of the experiment with ISIS-Tutor

Group	Number of steps	Concept repetitions	Task repetitions
A. Non-adaptive	81.33	11.17	6.17
B. Adaptive annotation	65.20	5.00	0.80
C. Adaptive annotation and hiding	58.20	4.80	0.40

Though not all the results of the experiment are processed by now, we can already report the results concerning our main hypothesis. These results are shown in the table 5 (all data are average numbers for each group). As we can see, the overall number of navigation steps, the number of unforced repetitions of previously studied concepts, and the number of task repetitions (i.e., trials to solve previoulsy visited task) are much less for both versions with adaptive navigation support. For the overall number of navigation steps and the number of task repetitions the difference was significant (we have used ANOVA to check the significance). On a paired test the significant difference for all three variables was found between non-adaptive group and joint adaptive group (B+C), but no significant difference was found between the two adaptive groups.

The results of the experiment with ISIS-Tutor show that both applied adaptive navigation support techniques - hiding and annotation - are efficient adaptation techniques. These techniques can improve user performance in hypermedia by significantly reducing navigation difficulty. Adaptive annotation and hiding in an educational context can reduce user's floundering in the hyperspace and make learning with hypermedia more goal-oriented. With these kinds of adaptive navigation support, the user can achieve the same result using a smaller number of navigation steps and visits to hypernodes. It is interesting to compare our results with the results of Kaplan et al. (1993), presented in Section 4.2. We have seen that adaptive presentation in hypermedia can reduce the time for learning the material and improve the comprehension of it, but cannot reduce the number of nodes visited in the process of learning. At the same time, adaptive annotation of links can hardly improve the quality of learning, but can reduce the number of visited nodes thus further reducing the learning time. These techniques look complimentary and can be used together for further improvement of the effectiveness of learning with hypermedia.

Conclusion

Adaptive hypermedia is a new direction of research at the crossroads of hypermedia, adaptive systems and intelligent systems. It is a promising and fast developing direction: more and more groups are starting work on adaptive hypermedia problems, more and more techniques are being developed. A weak point of current research on adaptive hypermedia is the luck of experimental investigation of AH efficiency. Very few reported AH techniques have been validated by empirical study, and only two of these studies reported significant results. However, the results of the reported studies are quite promising. It was found that adaptive presentation can significantly improve user comprehension of hypertext material, while adaptive navigation support techniques can significantly decrease user search and navigation efforts. Both group of techniques can decrease the time required to complete a task of the user. These groups of techniques can compliment each other and provide even better effect when used together.

Recently, adaptive hypermedia as a direction of research has received special attention in the WWW context We argue that adaptive hypermedia is one of the ways to increase the functionality of WWW. It can be demonstrated by several systems which implement adaptive hypermedia techniques in WWW context. We hope that the research works presented in this chapter can be used as a source of ideas for researchers interested in making local and networked hypermedia systems more efficient and functional.

Acknowledgments

Part of this work was supported by an Alexander von Humboldt-Stiftung Fellowship to the author.

References

Armstrong, R. et al. (1995) WebWatcher: A learning apprentice for the World Wide Web. *Proceedings of AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments.* Stanford, CA, March 27-29, 1995, http://www.isi.edu/sims/knoblock/sss95/mitchell.ps.

Beaumont, I. (1994) User modeling in the interactive anatomy tutoring system ANATOM-TUTOR. *User Modeling and User-Adapted Interaction* **4** (1), 21-45.

Boyle, C. and Encarnacion, A. O. (1994) MetaDoc: an adaptive hypertext reading system. *User Modeling and User-Adapted Interaction* **4** (1), 1-19.

Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction* **6** (2-3), 87-129.

Brusilovsky, P. and Pesin, L. (1994) ISIS-Tutor: An adaptive hypertext learning environment. *Proceedings of JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering*. Edited by H. Ueno and V. Stefanuk. Pereslavl-Zalesski, Russia, May 10-13, 1994, pp. 83-87.

Brusilovsky, P. and Pesin, L. (1995) Visual annotation of links in adaptive hypermedia. *Proceedings of CHI'95 (Conference Companion)*. Edited by I. Katz et al. Denver, May 7-11, 1995, pp. 222-223.

Brusilovsky, P. et al. (1996a) ELM-ART: An intelligent tutoring system on World Wide Web. In *Intelligent Tutoring Systems, Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96*, Lecture Notes in Computer Science, Vol. 1086, C. Frasson et al. (eds), Springer Verlag, Berlin. pp. 261-269.

Brusilovsky, P. et al. (1996b) A tool for developing adaptive electronic textbooks on WWW. *Proceedings of WebNet'97, World Conference of the Web Society*. San Francisco, CA, October 15-19, 1996, http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html.

Brusilovsky, P. and Zyryanov, M. (1993) Intelligent tutor, environment and manual for physical geography. *Proceedings of Seventh International PEG Conference*. Edinburgh, Edinburgh, 2-4 July, 1993, pp. 63-73.

Brusilovsky, P. L. (1992) Intelligent Tutor, Environment and Manual for Introductory Programming. *Educational and Training Technology International* **29** (1), 26-34.

Clibbon, K. (1995) Conceptually adapted hypertext for learning. *Proceedings of CHI'95 (Conference Companion)*. Edited by I. Katz et al. Denver, May 7-11, 1995, pp. 224-225.

de La Passardiere, B. and Dufresne, A. (1992) Adaptive navigational tools for educational hypermedia. In *Computer Assisted Learning, Proceedings of ICCAL'92, 4-th International Conference on Computers and Learning*, I. Tomek (ed) Springer-Verlag, Berlin. pp. 555-567.

de Rosis, F. et al. (1993) User tailored hypermedia explanations. *Proceedings of INTERCHI'93 (Adjunct proceedings)*. Amsterdam, 24-29 April 1993, pp. 169-170.

Debevc, M. et al. (1994) Adaptive bar implementation and ergonomics. *Informatica : Journal of Computing and Informatics* **18**, 357-366.

Espinoza, F. and Höök, K. (1996) An interactive WWW interface to an adaptive information system. *Proceedings of Workshop "User Modelling for Information Filtering on the World Wide Web" at the Fifth International Conference on User Modeling.* , http://www.sics.se/~kia/espinoza_hook.html.

Fischer, G. et al. (1990) Minimalist explanations in knowledge-based systems. *Proceedings of 23-th Annual Hawaii International Conference on System Sciences*. Kailua-Kona, HI, January 2-5, 1990, pp. 309-317.

Gonschorek, M. and Herzog, C. (1995) Using hypertext for an adaptive helpsystem in an intelligent tutoring system. *Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education*. Edited by J. Greer. Washington, DC, 16-19 August 1995, pp. 274-281.

Hohl, H. et al. (1996) Hypadapter: An adaptive hypertext system for exploratory learning and programming. *User Modeling and User-Adapted Interaction* **6** (2-3), 131-156.

Höök, K. (in press) Evaluating the usefulness of an adaptive hypermedia system. *Proceedings of 1997 International Conference on Intelligent User Interfaces*. Orlando, Florida, January 6-9 1997.

Höök, K. et al. (1996) A glass box approach to adaptive hypermedia. *User Modeling and User-Adapted Interaction* **6** (2-3), 157-184.

Kaplan, C. et al. (1993) Adaptive hypertext navigation based on user goals and context. *User Modeling and User-Adapted Interaction* **3** (3), 193-220.

Kaptelinin, V. (1993) Item recognition in menu selection: The effect of practice. *Proceedings of INTERCHI'93 (Adjunct Proceedings)*. Amsterdam, 24-29 April 1993, pp. 183-184.

Kay, J. and Kummerfeld, R. J. (1994) An individualised course for the C programming language. *Proceedings of Second International WWW Conference*. Chicago, IL, 17-20 October, http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.html.

Kobsa, A. et al. (1994) KN-AHS: An adaptive hypertext klient of the user modeling system BGP-MS. *Proceedings of Fourth International Conference on User Modeling*. Hyannis, MA, 15-19 August 1994, pp. 31-36.

Kushniruk, A. and Wang, H. (1994) A hypermedia-based educational system with knowledge-based guidance. *Proceedings of ED-MEDIA'94 - World conference on educational* *multimedia and hypermedia*. Edited by T.Ottman and I.Tomek. Vancouver, Canada, June 25-30, 1994, pp. 335-340.

Lai, M.-C. et al. (1995) Toward a new educational environment. *Proceedings of 4th International World Wide Web Conference*. , Boston, USA, December 11-14, 1995, http://www.w3.org/pub/Conferences/WWW4/Papers/238/.

Mallery, J. C. (1994) A Common LISP hypermedia server. *Proceedings of the First International Conference on the World-Wide Web*. , May 25, 1994.

Mathé, N. and Chen, J. (1996) User-centered indexing for adaptive information access. *User Modeling and User-Adapted Interaction* **6** (2-3), 225-261.

Milosavljevic, M. et al. (1996) Text Generation in a Dynamic Hypertex Environment. *Proceedings of Nineteenth Australasian Computer Science Conference (ACSC'96).* Melbourne, Australia, 31 January - 2 February 1996, pp. 417-426, http://www.mri.mq.edu.au/~mariam/papers/acsc/.

Moore, J. D. and Swartout, W. R. (1989) Pointing: A way toward explanation dialogue. *Proceedings of Eight National Conference on Artificial Intelligence.* , pp. 457-464.

Paris, C. L. (1988) Tailoring object description to a user's level of expertise. *Computational Linguistics* **14** (3), 64-78.

Pérez, T. et al. (1995) An adaptive hypermedia system. *Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education*. Edited by J. Greer. Washington, DC, 16-19 August 1995, pp. 351-358.

Vassileva, J. (1996) A task-centered approach for user modeling in a hypermedia office documentation system. *User Modeling and User-Adapted Interaction* **6** (2-3), 185-224.

Zeiliger, R. (1993) Adaptive testing: contribution of the SHIVA model. In *Item banking: Interactive testing and self-assessment*, NATO ASI Serie F, Vol. 112, D. Leclercq and J. Bruno (eds), Springer-Verlag, Berlin. pp. 54-65.

Zhao, Z. et al. (1993) Visualization of semantic relations in hypertext systems. *Proceedings of ED-MEDIA'93, World conference on educational multimedia and hypermedia.* Edited by T.Ottman and I.Tomek. Orlando, FL, pp. 556-564.