User Modeling and User Adapted Interaction, 1996, v 6, n 2-3, pp 87-129
(Special issue on adaptive hypertext and hypermedia)

Methods and techniques of adaptive hypermedia

Peter Brusilovsky

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

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 used now in several application areas where the hyperspace is reasonably large and where a hypermedia application is expected to be used by individuals with different goals, knowledge and backgrounds. This paper is a review of existing work on adaptive hypermedia. The paper is centered around a set of identified methods and techniques of AH. It introduces several dimensions of classification of AH systems, methods and techniques and describes the most important of them.

Keywords: adaptive hypermedia, navigation support, collaborative user modeling, adaptive text presentation, intelligent tutoring systems, student models.

1 Introduction

Hypermedia systems have become increasingly popular in the last five years as tools for user-driven access to information. Adaptive hypermedia is a new direction of research within the area of user-adaptive systems. The goal of this research is to increase the functionality of hypermedia by making it personalized. Adaptive hypermedia (AH) systems build a model of the goals, preferences and knowledge of the individual user and use this throughout the interaction for adaptation to the needs of that user.

AH systems can be useful in any application area where 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 and knowledge may be interested in different pieces of information presented on a hypermedia page and may use different links for navigation. AH tries to overcome this problem by using knowledge represented in the user model to adapt the information and links being presented to the given user. Adaptation can also assist the user in a navigational sense, which is particularly relevant for a large hyperspace. Knowing user goals and knowledge, AH systems can support users in their navigation by limiting browsing space, suggesting most relevant links to follow, or providing adaptive comments to visible links. The goal of this paper is to provide an overview of recent work on the development of adaptive hypermedia systems.

Since this area of research is very new, the concept of adaptive hypermedia systems has not been clearly defined yet. To make the scope of the review more clear we use in this paper the following working definition:

> by adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user.

In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system, it should have a user model, and it should be able to adapt the hypermedia using this model (i.e. the same system can look different to the users with different models). We have identified more than 20 systems which can be named as adaptive hypermedia systems according to our criteria (Appendix 1). The analysis of these systems is the main content of our review. Note that not all known systems which are named or referred to as adaptive hypermedia satisfy our definition. Some of them are not full-fledged hypermedia systems (Brusilovsky, 1992b; Yetim, 1993; André & Rist, 1996); some of them are not really adaptive, but rather adaptable (Waterworth, 1996) (this distinction will be made clearer later). There are also some projects which suggest interesting relevant ideas but have not yet reached the implementation stage (Tomek, Maurer & Nassar, 1993; Zyryanov, 1996). All these works, however, contain interesting ideas and we refer to them when it is relevant to the main line of presentation.

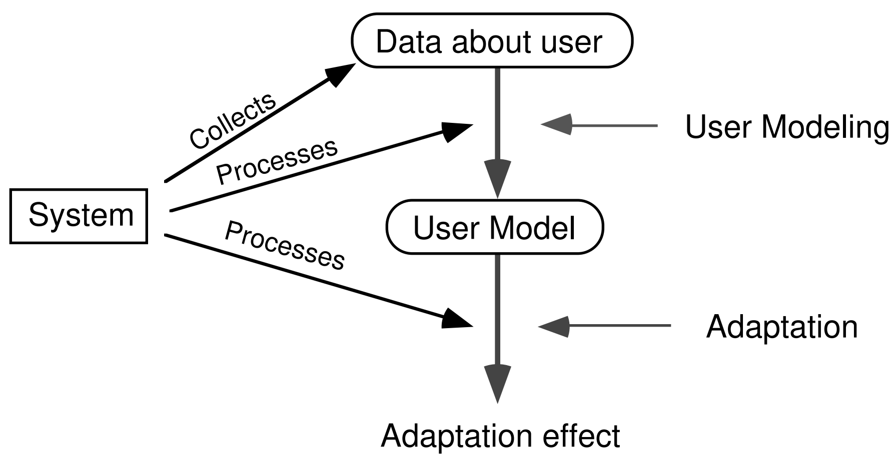

Figure 1. Classic loop "user modeling - adaptation" in adaptive systems

In this paper, the critical feature of adaptive hypermedia systems is the possibility of providing hypermedia adaptation on the basis of the user model. Therefore, the paper is centered around the problems of *adaptation*, the second part of the overall adaptation process in adaptive computer systems (Figure 1). The main content of the paper (sections 2-6) is a review of existing *methods and techniques* of adaptation in AH systems. The problems of *user modeling*, i.e. building and updating the user model in AH systems, are not a focus of the paper because they are not as critical for AH systems as a subclass of adaptive computer systems. Specific problems of user modeling in AH systems are discussed in section 7 which provides a comparative review of several methods of user modeling in AH systems. Special attention is paid to collaborative user modeling which are especially important for this AH system. The conclusion summarizes the content of the paper and discusses the prospects for research in the area of adaptive hypermedia.

2 Methods and techniques of adaptive hypermedia

Adaptation techniques refers to methods of providing adaptation in existing AH systems. These techniques are a part of the implementation level of an AH system. Each technique can be characterized by a specific kind of knowledge representation and by a specific adaptation algorithm. Adaptive hypermedia is a new area of research and most of the adaptation techniques are still unique in the sense that each of them was suggested in conjunction with the development of an AH system. However, some popular techniques were already implemented with minor variants in several earlier systems.

Adaptation methods are defined as generalizations of existing adaptation techniques. Each method is based on a clear adaptation idea which can be presented at the conceptual level. For example, "...insert the comparison of the current concept with another concept if this other concept is already known to the user", or "...hide the links to the concepts which are not yet ready to be learned". The same conceptual method can be implemented by different techniques. At the same time, some techniques are used to implement several methods using the same knowledge representation.

The set of methods and techniques forms a tool kit or an "arsenal" of adaptive hypermedia and can be used as a source of ideas for the designers and developers of adaptive hypermedia systems.

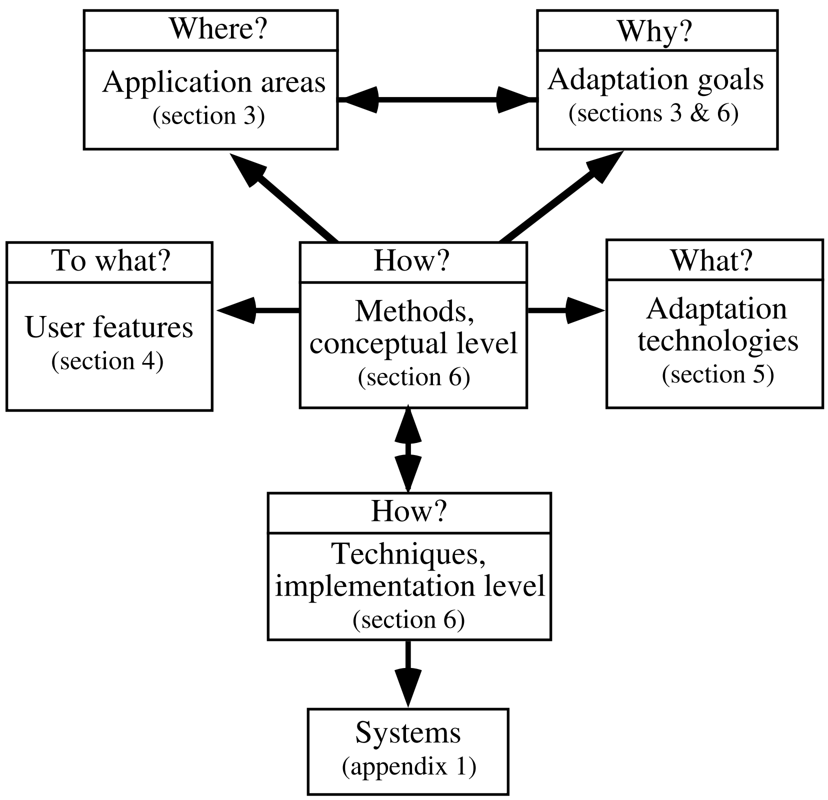

Figure 2. Possible classifications for AH methods and techniques.
An arrow stands for 1-to-N relationship.

To review AH systems it is first necessary to establish the basis for the classification of adaptive hypermedia methods and techniques (Figure 2). The identified dimensions are quite typical for the analysis of adaptive systems in general (Dieterich et al., 1993).

- The first dimension considered is *where adaptive hypermedia systems can be helpful*. The review identifies several application areas for AH systems (see Table 1) and for each area points the problems which can be partly solved by applying adaptive hypermedia techniques (section 3).
- The second dimension is *what features of the user are used as a source of the adaptation*, i.e. to what features of the user the system can adapt its behavior. The review identifies several

user features which are considered important by existing AH systems and discusses the common ways to represent them (section 4).

- The third dimension is *what can be adapted* by a particular technique. Which features of the system can be different for different users. Along this dimension the review identifies seven ways to adapt hypermedia (see Figure 4). They can be divided into two essentially different groups - content adaptation and link adaptation (section 5). I call different ways to adapt hypermedia *technologies of adaptation*.
- The fourth dimension of classification is the *adaptation goals* achieved by different methods and techniques: why these methods and techniques are applied, and which problems of the users they can solve. The adaptation goals are dependent on application areas. Each application area has its own set of problems and each goal is important in some range of application areas (section 3). The adaptation goals are considered in parallel with reviewing of relevant adaptation methods and techniques which implement these methods (section 6).

The four identified dimensions are very suitable to classify various application methods. Usually, each method is an application of a particular adaptation technology (such as text adaptation or hiding of links) to achieve one of possible adaptation goals using one of the users features as a source for adaptation (As an exception, methods can achieve more than one goal or use more than one feature of the user). According to its goals and used features a particular method can be useful in a subset of application areas.

3 Where and why adaptive hypermedia can be helpful

Unlike other kinds of application systems, any hypermedia system is adaptive in some sense: using free browsing different users can adapt the system to their information needs. Many researchers hold that it is the user who should bring the adaptivity to the man-machine hypermedia system. Why do we need any other kind of adaptation? Why do we need that a hypermedia system adapts itself to the particular user? The answer depends on an application area perspective. Analysis of existing AH systems allow us to name six kinds of hypermedia systems which are used at present as application areas in most of research projects on adaptive hypermedia. These are: educational hypermedia, on-line information systems, on-line help systems, information retrieval hypermedia systems, institutional information systems, and systems for managing personalized views (Table I). In each of these areas adaptive hypermedia techniques can be helpful because they help solve the identified problems. This section characterizes all these application areas, pointing out their specific features and identifying problems.

Educational Hypermedia Systems	Anatom-Tutor, C-Book, <Clibbon>, ELM-ART, ISIS-Tutor, ITEM/PG, HyperTutor, Land Use Tutor, Manuel Excel, SHIVA, SYPROS, ELM-PE, *Hypadapter, HYPERCASE*
On-line Information Systems	Hypadapter, HYPERCASE, KN-AHS, MetaDoc, PUSH, *HYPERFLEX, CID, Adaptive HyperMan*
On-line Help Systems	EPIAIM, HyPLAN, Lisp-Critic, ORIMUHS, WING-MIT, *SYPROS*
Information Retrieval Hypermedia	CID, DHS, Adaptive HyperMan, HYPERFLEX, WebWatcher
Institutional Hypermedia	Hynecosum
Personalized Views	Basar, Information Islands

Table I. Existing adaptive hypermedia systems classified according their application areas. Second entries for the systems that fit two categories are shown in *italics*.
Bibliographic references are provided in Appendix 1.

The most popular area for adaptive hypermedia research is educational hypermedia. Existing educational hypermedia systems have relatively small hyperspaces representing a particular course or section of learning material on a particular subject. The goal of the student is usually to learn all this material or a reasonable part of it. The hypermedia form supports student-driven acquisition of the learning material. The most important user feature in educational hypermedia is user knowledge of the subject being taught. Adaptive hypermedia techniques can be useful to solve a number of the problems associated with the use of educational hypermedia. Firstly, the knowledge of different users can vary greatly and the knowledge of a particular user can grow quite fast. The same page can be unclear for a novice and at the same time trivial and boring for an advanced learner. Second, novices enter the hyperspace of educational material knowing almost nothing about the subject. Most of the offered links from any node lead to the material which is completely new for them. They need navigational help to find their way through the hyperspace. Without such a help they can "get lost" even in reasonably small hyperspaces, or use very inefficient browsing strategies (Hammond, 1989).

Another popular application for adaptive hypermedia is the area of various *on-line information systems* from on-line documentation to electronic encyclopedias. The goal of these systems is to provide reference access to information (rather then a systematic introduction as in educational hypermedia) for the users with different knowledge level of the subject. Each node of the hyperspace usually represents one concept of the subject and contains several pages of information. Depending on the subject, the size of the hyperspace can range from reasonably small to very large. Similar to educational hypermedia, on-line information systems have problems with satisfying the needs of very different users. Those with different knowledge and background need different information about a concept and at different levels of detail. They usually have no time to browse all the information about the concept to look for the required portion of information. Users also have different goals when accessing an information system. In some cases they know which concepts to access to achieve their goals and do not need any navigational support (Boyle & Encarnacion, 1994; Kobsa, Müller & Nill, 1994). However, when the goal can not be directly mapped to the structure of the hyperspace or when the hyperspace is large, users need help in navigation and in finding relevant pieces of information. To provide such help, the system has to know the user's goal (Höök et al., 1996; Micarelli & Sciarrone, 1996). As we will see later (section 7) inferring the user's goal is a difficult problem in on-line information systems unless the goal is provided directly by the user (Höök et al., 1996).

Very close to on-line information systems are *on-line help systems*. These systems serve on-line information about computer applications (such as a spreadsheet, programming environment, or expert system) which is required to help the users this system. The difference from the former category is that on-line help systems are not independent as on-line information systems but are attached to their application system. Another difference is that the hyperspace in existing on-line help systems is reasonably small. As we will see later, the distinction between small and large hyperspace is important from adaptation point of view, and that gives a reason to distinguish these application areas. On-line help systems and on-line information systems share the problem of serving different information to different users. At the same time, the problem of helping users to find relevant pieces of information is less important for on-line help systems because the hyperspace is not large and because the system knows the context from which the user called for on-line help (context-sensitive help). The context of work in an application system provides a reliable source of information for an adaptive on-line help system to determine the user's goal and to offer the most relevant help items (Encarnação, 1995b; Grunst, 1993; Kim, 1995).

The three application areas listed above belong to traditional application areas for hypermedia. The majority of existing hypermedia systems belong to one of these three areas. It is not surprising that most adaptive hypermedia systems also belong to these areas. The three areas listed below are more recent application areas for hypermedia.

Information retrieval (IR) hypermedia systems is a new class of IR systems which combine traditional information retrieval techniques with a hypertext-like access from the index terms to documents and provide the possibility of browsing the hyperspace of documents using similarity links between documents (Agosti, Melucci & Crestani, 1995; Helmes, Razum & Barth, 1995). It is known that browsing can help users to find the required documents when they have problems with constructing a proper formal query. The size of hyperspace in regular IR hypermedia is usually very large and can not be structured "by hand". It means that the links in this hyperspace are not provided by a designer as in on-line information systems, but are calculated by the system (for example, using similarity measurements). Another difference from on-line information systems is that users of IR hypermedia are more often professionals in different areas who use the system in their everyday work with different IR goals. A very new special kind of IR hypermedia is IR systems on Word Wide Web which have slightly different nature in their links and a potentially unlimited hyperspace. Existing adaptive IR systems (Kok, 1991) show some ways to help the user in pure IR settings. Adaptive IR hypermedia systems can offer some additional help by limiting the navigation choice (Boy, 1991; Mathé & Chen, 1996) and by suggesting the most relevant links to follow (Armstrong et al., 1995; Kaplan, Fenwick & Chen, 1993; Katsumoto, Fukuda & Shibata, 1996; Mathé & Chen, 1996).

Another new area of application for adaptive hypermedia is *institutional information systems* which serves on-line all the information required to support the work of some institution, for example, a hospital (Vassileva, 1996). Originally, these kinds of systems were developed as a set of loosely related databases, but in some recent systems such databases are joined into a single hyperspace which can be reasonably large. A specific feature of these systems is that they are a medium for everyday work of many institution employees. According to their profession they may always use only a specific area of hyperspace, and according to the current working goal they may need an access to a very small subset of it. Most of users never need to access the parts of the hyperspace outside their working area, moreover, too many navigation opportunities somewhat distract them from their primary work. In this respect *work-oriented* institutional information systems significantly differs from *search-oriented* IR hypermedia and on-line information systems where the "working area" of a user is the entire hyperspace. At the same time users of institutional information systems may need assistance in organizing a more convenient personalized access to their working areas (Vassileva, 1996). Another problem of institutional information systems which is similar to one of the problems of educational hypermedia is related to new employees who are not familiar with the structure of the hyperspace (though they can be familiar with the application domain itself) and can get lost even in their small professional subarea.

The last of the new application areas is *systems for managing personalized views in information spaces* such as Information Islands (Waterworth, 1996) and Basar (Thomas, 1995). Existing telecommunication systems such as World-Wide Web (WWW) offer huge amount of different information and on-line services which form a really unlimited hyperspace. Many users need to have an access too one or more subsets of all the hyperspace for their everyday work. To protect themselves from the complexity of the overall hyperspace, they maybe interested to define *personalized views* on the entire hyperspace. Each view can be devoted to one of the goals or interests related with the work of the user. Partly, this application area is similar to institutional hypermedia and other kinds of information systems where users need a convenient access to a subset of an information space for everyday work. A new factor which affects systems dealing with wide (and *world-wide*) information spaces is the dynamic character of hyperspace where items can appear, disappear, or evolve. Personalized views in world-wide information spaces require permanent management: searching for new and relevant items and identifying expired or changed items (in this sense, this application area is similar to IR hypermedia). Adaptation to the user goals, interests, and background can help to solve the identified problems (Thomas, 1995; Thomas & Fischer, 1996).

It is worthwhile to stress again that all six listed application areas are not mutually exclusive. Some of them are pairwise similar and share the same problems. These pairs are: IR hypermedia and on-line information systems, on-line information/help systems and educational hypermedia, educational hypermedia and institutional hypermedia, institutional hypermedia and information space management systems. Also, the difference between neighboring areas is not always clear-cut and some systems belong to both areas, for example, Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996) and HYPERCASE (Micarelli & Sciarrone, 1996) share features of educational hypermedia and on-line information systems, and HYPERFLEX shares features of on-line information systems and IR hypermedia. In fact, all mentioned application areas can be ordered along a continuum (Figure 3) where similar areas are placed together. This diagram offers us some interesting insights to the order and structure of hypermedia.

More traditional educational hypermedia and on-line help are located in the center. These systems are "real hypermedia," and can demonstrate all classic hypermedia features: all kinds of links including contextual links, indexes, local and global maps, guided tours, etc. The systems located at the ends of the continuum use only some of the traditional hypermedia features. The size of the hyperspace grows from reasonably small in the center to huge on both ends. It is not surprising that the systems standing on both the ends of the continuum are also similar, because they share the problems related with large hyperspace. The systems above the center are more search-oriented while the systems below are more work-oriented. According to that, the relative size of *working subset* decreases gradually from top to bottom from the entire hyperspace in IR hypermedia to a very small personalized subset of it in Basar-like systems.

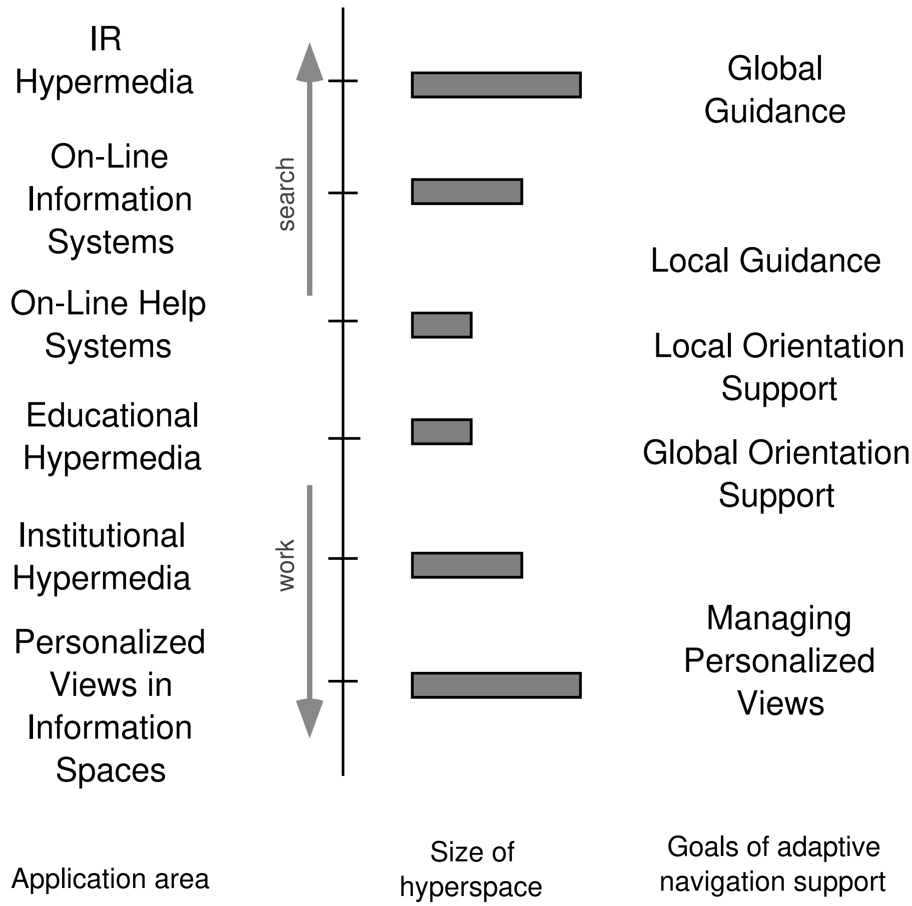

Figure 3. The continuum of adaptive hypermedia systems

4 Adapting to what?

The second question to pose when speaking about a particular kind of adaptive system is: What aspects of the user working with the system can be taken into account when providing adaptation? To which features – that can be different for different users (and may be different for the same user at different time) – can the system adapt? Generally, there are many features related to the current context of the user work and to the user as an individual which can betaken into account by an adaptive system. So far, this paper identifies five features which are used by existing adaptive hypermedia systems: users' goals, knowledge, background, hyperspace experience, and preferences.

4.1 Knowledge

User's knowledge of the subject represented in the hyperspace appears to be the most important feature of the user for existing adaptive hypermedia systems. It is used by about one third of adaptation techniques. Almost all adaptive presentation techniques rely on user's knowledge as a source of adaptation. User's knowledge is a variable for a particular user. This means that an adaptive hypermedia system which relies on user's knowledge has to recognize the changes in the user's knowledge state and update the user model accordingly.

User's knowledge of the subject is most often represented by an *overlay model* (Hypadapter, EPIAIM, KN-AHS, ITEM/PG, ISIS-Tutor, ELM-ART, SHIVA, HyperTutor) which is based on the structural model of the subject domain. Generally, the structural domain model is represented as a network of domain *concepts*. The concepts are related with each other thus forming a kind of semantic network which represents the structure of the subject domain. These concepts can be named differently in different systems - topics, knowledge elements, objects, learning outcomes - but in all the cases they are just elementary pieces of knowledge for the given domain. Some systems use a simplified form of the domain model (Boyle & Encarnacion, 1994; Zeiliger, 1993) without any links between concepts, but the majority of AH systems use a rather advanced domain models with several types of concepts which represent different kinds of knowledge elements or objects and several kinds of links which represent different kinds of relationships between concepts (Hypadapter, EPIAIM, PUSH, Anatom-Tutor, KN-AHS, ITEM/PG, ELM-ART, ITEM/IP, SHIVA, HyperTutor).

The idea of the overlay model is to represent an individual user's knowledge of the subject as an "overlay" of the domain model. For each domain model concept, an individual overlay model stores some value which is an estimation of the user knowledge level of this concept. This can be just a binary value (known–not known), a qualitative measure (good-average-poor), or a quantitative measure, such as a probability that the user knows the concept. An overlay model of user knowledge can be represented as a set of pairs "concept - value", one pair for each domain concept. Overlay models are powerful and flexible, they can independently measure user knowledge of different topics. Overlay models were originally developed in the area of intelligent tutoring systems and *student* modeling (Greer & McCalla, 1993). In many ITS the student model is just the overlay model of student knowledge. As a result, in the area of adaptive interfaces, an overlay model of user's knowledge (which is a part of the overall user model) is sometimes called the student model (Benyon & Murray, 1993).

Sometimes a simpler stereotype user model is used to represent the user's knowledge (Beaumont, 1994; Boyle & Encarnacion, 1994; Hohl, Böcker & Gunzenhäuser, 1996). A stereotype user model distinguishes several typical or "stereotype" users. For each dimension of user modeling the system can have a set of possible stereotypes. For example, MetaDoc uses two dimensions of classification and two sets of stereotypes (novice - beginner - intermediate - expert): one to represent user's knowledge of general computer concepts, another to represent user's knowledge of UNIX (which is the domain of the system). A particular user is usually modelled by assigning this user to one of stereotypes for each dimension of classification (for example, intermediate for general computer concepts, novice for UNIX).

A stereotype user model can also be represented as a set of pairs or "stereotype-value", where the value can be not only "true" or "false" (what means that the user belongs or does not belong to the stereotype) but also some probabilistic value (what represents the probability that the user belongs to the stereotype). Stereotype model is simpler and less powerful then overlay model but it is also more general and much easier to initialize and to maintain.

A problem with the stereotype model of knowledge is that many efficient adaptation techniques require a more fine-grained overlay model. One way to solve this problem is to provide a mapping from stereotype to overlay model. This can be done by associating a fixed set of pairs concept-value with each stereotype (Boyle & Encarnacion, 1994; de Rosis, De Carolis & Pizzutilo, 1993), or by a more flexible way which is based on the "difficulty" of concepts (Hohl et al., 1996). In turn, an overlay user model has the problem of initialization - it is very hard to set all its values after a short interview with a new user. Good results can be achieved by combining stereotype and overlay modeling (Anatom-Tutor, EPIAIM, KN-AHS, Hypadapter). They can be combined in the following way (de Rosis et al., 1993; Hohl et al., 1996): stereotype modeling is used at the beginning of work to classify a new user and to set initial values for overlay model, then a regular overlay model is used.

4.2 Goals

User's goal or user's task is a feature related with the context of a user's work in hypermedia rather than with the user as an individual. Depending on the kind of system, it can be the goal of the work (in application systems), a search goal (in information retrieval systems), and a problem-solving or learning goal (in educational systems). In all of these cases the goal is an answer to the question "Why is the user using the hypermedia system and what does the user actually want to achieve?" User's goal is the most changeable user feature: almost always it changes from session to session and often can change several times within one session of work. In some systems it is reasonable to distinguish local or low-level goals which can change quite often and general or high level goals and tasks which are more stable. For example, in educational systems the learning goal is a high-level goal, while the problem-solving goal is a low-level goal which changes from one educational problem to another several times within a session. The user's goal can be considered as a very important feature of the user for adaptive hypermedia systems. Almost one third of existing adaptation techniques rely on it. Interestingly, nearly all of these techniques are adaptive navigation support techniques.

The user current goal is usually modeled by a way which is somewhat similar to overlay knowledge modeling. As a rule, each system supports a set of possible user goals or tasks which it can recognize (HyPLAN, ORIMUHS, PUSH, HYPERCASE, Hynecosum, HYPERFLEX). In some cases, the set of goals is very small and the goals are not related to each other (Höök et al., 1996; Kaplan, Fenwick & Chen, 1993). To model the current user goal, the system includes one of these goals into the user model. More advanced goal-based systems (Encarnação, 1995; Grunst, 1993; Vassileva, 1996) use a more advanced representation of possible goals and current user goals. The most advanced representation of possible user goals is a hierarchy (a tree) of tasks (Vassileva, 1996). The most advanced representation of user current goals is a set of pairs "goal-value" where the value is usually the probability that the corresponding goal is the current goal of the user (Encarnação, 1995; Grunst, 1993; Micarelli & Sciarrone, 1996).

4.3 Background and Experience

Two features of the user which are similar to user's knowledge of the subject but functionally differ from it are user's background and user's experience in the given hyperspace. By *user's background* we mean all the information related to the user's previous experience outside the subject of the hypermedia system, which is relevant enough to be considered. This includes the user's profession, experience of work in related areas, as well as the user's point of view and perspective. The systems EPIAIM, C-Book, and Anatom-Tutor include user's background feature in the user model and apply it to adaptive presentation and Adaptive HyperMan applies it to adaptive navigation support.

By *user's experience* in the given hyperspace we mean how familiar is the user with the structure of the hyperspace and how easy can the user navigate in it. This is not the same as user's knowledge of the subject (Vassileva, 1996). Sometimes, the user who is generally quite familiar with the subject itself is not familiar at all with the hyperspace structure. Vice versa, the user can be quite familiar with the structure of the hyperspace without deep knowledge of the subject. One more reason to distinguish hyperspace experience from knowledge level is the existence of an adaptive navigation technique (Pérez, Gutiérrez & Lopistéguy, 1995; Vassileva, 1996) which relies on this feature of the user (see section 6.4).

Such individual features of a user as background or experience are usually also modeled by a stereotype user model (MetaDoc, Anatom-Tutor, EPIAIM, C-Book). The stereotype can be an experience stereotype (Pérez et al., 1995; Vassileva, 1996), or a background stereotype for such dimensions as profession (de Rosis et al., 1993), prospect (Beaumont, 1994), or native language (Kay & Kummerfeld, 1994).

4.4 Preferences

The last, but not the least important feature of the user considered by adaptive hypermedia systems is *user's preferences*. For different reasons the user can prefer some nodes and links over others and some parts of a page over others. These preferences can be absolute (Hypadapter, Information Islands) or relative, i.e., dependent from the current node, goal (PUSH, HYPERFLEX) and current context in general (WebWatcher, CID, DHS, Adaptive HyperMan). Preferences are used most heavily in IR hypermedia and, at the same time, in most adaptive IR hypermedia systems preferences are the only stored information about the user.

User's preferences differ from other user model components in several aspects. Unlike other components, the preferences can not be deduced by the system. The user has to inform the system directly or indirectly (by a simple feedback) about such preferences. It looks more close to adaptability then to adaptivity. The difference is that adaptive hypermedia systems can generalize the user's preferences and apply it for adaptation in new contexts (Armstrong et al., 1995; Boy, 1991; Höök et al., 1996; Kaplan et al., 1993; Katsumoto et al., 1996; Mathé & Chen, 1996). Another specific feature of preference modeling is the way of representation. While other parts of the user model are usually represented symbolically, preferences are often represented and calculated numerically by very special ways (Kaplan et al., 1993; Katsumoto et al., 1996; Mathé & Chen, 1996). The numeric way of presentation has some preferences over the symbolic way: it opens the possibility to combine several user models and to accumulate a *group user model* (Kaplan et al., 1993; Mathé & Chen, 1996). Group models accumulate preferences of a specific group of users (such as a research laboratory). A group model is a nice starting model for a new member of the group. Group models are important also for collaborative work (it is very hard to collaborate when collaborators use individual user models and thus have different adapted views on the same subject).

5 What can be adapted in adaptive hypermedia?

An important question to ask when speaking about any kind of adaptive systems is: what can be adapted in this system? Which features of the system can differ for different users? What is the space of possible adaptations? In adaptive hypermedia, the adaptation space is quite limited: there are not so many features which can be altered. At some level of generalization, hypermedia consist of a set of nodes or *hyperdocuments* (for the purpose of brevity we will call them "pages") connected by links. Each page contains some local information and a number of links to related pages. 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). We distinguish content-level and link-level adaptation as two different classes of hypermedia adaptation and call the first one adaptive presentation and the second one adaptive navigation support (Figure 4).

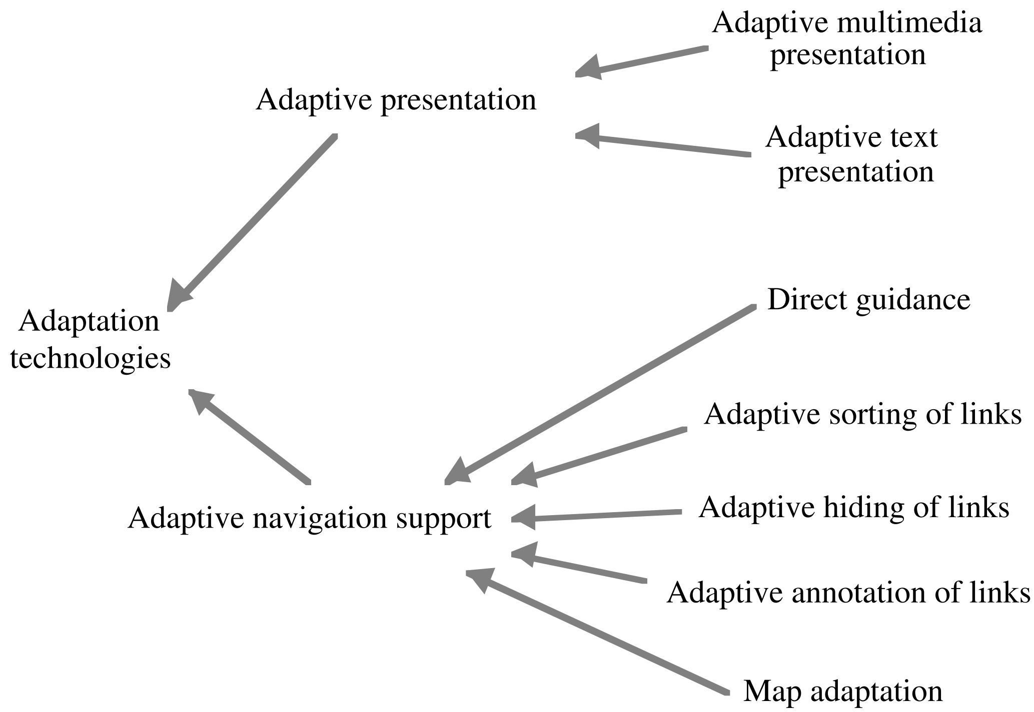

Figure 4. Adaptation technologies in adaptive hypermedia

5.1 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. In hypermedia systems, the content of a regular page may be not only a text as in classic hypertext systems but also a set of various multimedia items. From this point of view, we could distinguish adaptive text presentation and adaptive multimedia presentation in hypermedia systems. However, at present all work on adaptive presentation in hypermedia are really the works on adaptive text presentation. Some existing adaptive hypermedia systems do contain non-textual items (Brusilovsky & Zyryanov, 1993; de Rosis et al., 1993; Kobsa et al., 1994), but can not present these items adaptively. At the same time, there are a number of good techniques for adaptive multimedia presentation (Maybury, 1993; André & Rist, 1996) but these techniques have never been used in full-fledged hypermedia systems.

As for adaptive text presentation, it 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; Böcker et al., 1990; Boyle & Encarnacion, 1994; Brusilovsky, 1992b; 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 & Swartout, 1989; Paris, 1988; Zukerman & McConachy, 1993). As we will show in the following sections, there are a number of different techniques for adaptive text presentation. We group these techniques into one single technology because they look very similar from a "what can be adapted" point of view: users with different user models get different texts as a content of the same page.

5.2 Adaptive navigation support

The idea of adaptive navigation support techniques is to help users to find their paths in hyperspace by adapting the way of presenting links to goals, knowledge, and other characteristics of an individual user. Though this area of research is very new, a number of interesting techniques have been already suggested and implemented. These techniques can be classified in five groups according to the way they use to adapt presentation of links. We distinguish five technologies for adapting link presentation which are different from a "what can be adapted" point of view: direct guidance, sorting, hiding, annotation, and map adaptation (Figure 4). To compare these technologies we first need to understand how and in which context the links are usually presented. Here we mean links in the user's sense (i.e., visible and "clickable" representation of the related pages to which the user can navigate). We distinguish four kinds of link presentation which are different from the point what can be altered and adapted:

Local non-contextual links. This type includes all kinds of links on regular hypermedia pages which are independent from the content of the page. They can appear as a set of buttons, a list, or a pop-up menu. These links are easy to manipulate - they can be sorted, hidden, or annotated.

Contextual links or "real hypertext" links. This type comprises "hotwords" in texts, "hot spots" in pictures, and other kinds of links which are embedded in the context of the page content and can not be removed from it. These links can be annotated but can not be sorted or completely hidden.

Links from index and content pages. An index or a content page can be considered as a special kind of page which contains only links. These links are usually presented in a fixed order (content order for content pages and alphabetic order for index pages). As a rule, links from index and content pages are non-contextual, unless such a page is implemented in a form of a picture.

Links on local maps and links on global hyperspace maps. Maps usually graphically represent a hyperspace or a local area of hyperspace as a network of nodes connected by arrows. Using maps, the user can directly navigate to all nodes visible on the map just by clicking on a representation of the desired node. From a navigation point of view, these clickable representations of nodes are exactly what we mean above by links, while the arrows serving as a representation of links are not used for direct navigation.

Now we can compare existing technologies of link adaptation. *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 & Pesin, 1994), SHIVA (Zeiliger, 1993), HyperTutor (Pérez et al., 1995), and Land Use Tutor (Kushniruk & Wang, 1994). The former way is more clear, while the latter is more flexible, because it can be used to recommend the node which is not connected directly to the current one (and not represented on the current page). Direct guidance is a clear and easy to implement technology, it can be used with all four kinds of link presentation listed above. The problem with direct guidance is that it provides limited support: "follow me or no help". Direct guidance can hardly be the primary form of navigation support because it provides no support for the users who would not like to follow the system's suggestion. Direct guidance is useful but it has to 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 more close 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 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. At the same time, some recent research shows that the stable order of options in menus is important for novices (Debevc, Rajko & Donlagic, 1994; Kaptelinin, 1993). However, this technology seems to be useful for information retrieval (IR) applications (Armstrong et al., 1995; Kaplan et al., 1993; Mathé & Chen, 1996). 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 similar area where adaptive ordering can be used is on-line documentation systems (Hohl, Böcker & Gunzenhäuser, 1996). There are also some suggestions on using adaptive ordering in educational hypermedia (Tomek et al., 1993).

Hiding is currently the most often used technology for adaptive navigation support. The idea of navigation support by hiding is to restrict the navigation space by hiding links to "not relevant" pages. The page can be considered as not relevant for several reasons: for example, if it is not related to the user's current goal (Boy, 1991; Brusilovsky & Pesin, 1994; Grunst, 1993; Höök et al., 1996; Vassileva, 1996) or if it presents materials which the user is not prepared yet to understand (Brusilovsky & Pesin, 1994; Clibbon, 1995; Gonschorek & Herzog, 1995; Pérez et al., 1995). From the surface view, hiding looks like the most obvious and the most easy to implement technology. It protects users from the complexity of the unrestricted hyperspace and reduces their cognitive overload. Hiding has a wide applicability: it can be used with all kinds of non-contextual, index, and map links by real hiding buttons or menu items (Brusilovsky & Pesin, 1994), and with contextual links by transferring clickable "hot words" to normal text (Gonschorek & Herzog, 1995; Pérez et al., 1995). Hiding is also more transparent to the user and looks more "stable" for them than adaptive ordering (links are usually added incrementally, but not removed or reordered).

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 or in the form of visual cues using, for example, different icons (de La Passardiere & Dufresne, 1992; Schwarz et al., 1996), colors (Brusilovsky & Pesin, 1994; Brusilovsky & Zyryanov, 1993), or font sizes (Hohl, Böcker & Gunzenhäuser, 1996). A good review of various visual cues which can be used in hypermedia systems can be found in (Arens & Hammwöhner, 1995). Link annotation is known as an effective technology 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 & Pesin, 1994; Schwarz et al., 1996) can distinguish and annotate differently up to six states on the basis of the user model (see section 6.4)

Annotation seems to be a very relevant form of adaptive navigation support. Annotation can be naturally used with all four 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 nodes - relevant and non relevant - while annotation, as mentioned above, up to six states, in particular, several levels of relevancy as it implemented in Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996). Annotations do not restrict cognitive overload as much as hiding does, but the 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), but dimmed links are still visible (and traversable, if required) which protects the user from forming wrong mental maps.

Map adaptation technology comprises various ways of adapting the form of global and local hypermedia maps presented to the user. Such technologies as direct guidance, hiding, and annotation also can be used to adapt hypermedia maps, but all these technologies do not change the form or the structure of maps. Human-computer interaction research offers a number of techniques to adapt the structure and the form of various kinds of networks, including hypermedia maps (Furnas, 1986; Mukherjea & Foley, 1995; Mukherjea, Foley & Hudson, 1995; Rivlin, Botafogo & Shneidermann, 1994). However, most of these ideas have not been applied yet in adaptive hypermedia systems. The only known exeption (see section 6.4) is the HYPERCASE system (Micarelli & Sciarrone, 1996) which is partly based on the ideas of (Rivlin et al., 1994).

Direct guidance, sorting, hiding, annotating, and map adaptation 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 contradictory and can be used in combinations. For example, ISIS-Tutor (Brusilovsky & Pesin, 1994) uses direct guidance, hiding, and annotation and Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996) uses sorting, hiding, and annotation. In particular, the direct guidance technology can be naturally used in combination with any of the three other technologies.

6 How adaptive hypermedia can help

In this section we consider methods by which adaptive hypermedia systems can help to solve some hypermedia problems and describe the most interesting techniques applied by existing AH systems to implement these methods. Since content adaptation techniques and adaptive navigation support techniques are intended to solve different problems we consider them separately.

6.1 How content adaptation can help: methods

At present, content adaptation is used in three classic hypermedia application areas: on-line information systems - Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996), MetaDoc (Boyle & Encarnacion, 1994), KN-AHS (Kobsa et al., 1994), and PUSH (Höök et al., 1996), on-line help systems - Lisp-Critic (Fischer et al., 1990), EPIAIM (de Rosis et al., 1993), WING-MIT (Kim, 1995), and ORIMUHS (Encarnação, 1995b), and educational hypermedia - Anatom-Tutor (Beaumont, 1994), ITEM/IP (Brusilovsky, 1992b), C-book (Kay & Kummerfeld, 1994b), and SYPROS (Gonschorek & Herzog, 1995) (Table II).

The goal of the most popular method of content adaptation (we call it *additional explanations*) is to hide from the user some parts of information about a particular concept which are not relevant to the user's level of knowledge about this concept. For example, low level details can be hidden from users a with poor level of knowledge of this concept because they can not understand these details. On the contrary, additional explanations usually required by novices to understand the concept can be hidden from a user with a good level of knowledge of the concept because they do not need these explanations anymore. In more general terms, in addition to the basic presentation, some category of users can get some additional information which is specially prepared for this category of users and will not be shown to users of other categories. This method is used in MetaDoc (Boyle & Encarnacion, 1994), KN-AHS (Kobsa et al., 1994), ITEM/IP (Brusilovsky, 1992b), EPIAIM (de Rosis et al., 1993), and Anatom-Tutor (Beaumont, 1994). A goal-based variant of this method is to hide from the user some parts of information about a particular concept which are not relevant to the current user's goal (Höök et al., 1996).

Two other methods *prerequisite explanations* and *comparative explanations* change the information presented about a concept depending on the user knowledge level of related concepts. The first method is based on prerequisite links between concepts. The idea is the following: before presenting an explanation of a concept the system inserts explanations of all its prerequisite concepts which are not sufficiently known to the user. This method is used in Lisp-Critic (Fischer et al., 1990) and C-book (Kay & Kummerfeld, 1994b). The second method is based on similarity links between concepts. If a concept similar to the concept being presented is known, the user gets a comparative explanation which stress similarities and differences between the current concept and the related one. Such comparative explanations are particularly effective in the domain of programming languages. Interestingly, all systems which apply this method (ITEM/IP, Lisp-Critic, and C-book) belong to this domain.

Another method (we call it *explanation variants*) assumes that showing or hiding some portion of the content is not always sufficient for the adaptation because different users may need essentially different information. With this method, the system stores several variants for some parts of the page content and the user gets the variant which corresponds to his or her user model. This method is used in Anatom-Tutor (Beaumont, 1994), Lisp-Critic (Fischer et al., 1990), Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996), ORIMUHS (Encarnação, 1995b), SYPROS (Gonschorek & Herzog, 1995), and WING-MIT (Kim, 1995).

An interesting method which can take into account both the user's background and knowledge level is *sorting* the fragments of information about the concept where the information which is most relevant to user's background and knowledge is placed toward the front. This method is implemented in Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996) and EPIAIM (de Rosis et al., 1993).

	Conditional text	Stretchtext	Fragment variants	Page variants	Frame-based technique
Additional, prerequisite, and comparative explanations	C-Book ITEM/IP Lisp-Critic	MetaDoc KN-AHS PUSH			EPIAIM PUSH
Explanation variants	C-Book		Anatom-Tutor Lisp-Critic WING-MIT	Anatom-Tutor C-Book EPIAIM ORIMUHS SYPROS	Hypadapter
Sorting					EPIAIM Hypadapter

Table II. Adaptive presentation: methods, techniques, and systems

6.2 How content adaptation can help: techniques

A simple, but effective technique for content adaptation is the *conditional text* technique which is used in ITEM/IP (Brusilovsky, 1992b), Lisp-Critic (Fischer et al., 1990), and C-book (Kay & Kummerfeld, 1994b). With this technique, all possible 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 all the methods of adaptation listed above excluding sorting. A simple example is hiding chunks with irrelevant explanations if the user's knowledge level of the current concept is good enough, or turning on a chunk with comparative explanations if the corresponding related concept is already known.

A more high level technique which can also turn off and on different parts of the content according to the user knowledge level is suggested in the MetaDoc (Boyle & 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 activation of a hot word is moving to another page with related text. In stretchtext this related text can simply replace the activated hotword (or a phrase with this hotword) extending the text of the current page. If required, this extended or "uncollapsed" text may be collapsed back to a hotword. Each node in MetaDoc is a stretchtext page which may contain many "uncollapsable" hotwords. The idea of adaptive stretchtext presentation in MetaDoc is to present a requested page with all stretchtext extensions 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 a high degree of knowledge of a concept will always get 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 get 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 and that it can take into account both the knowledge and the preferences of the user. After optional presentation of the stretchtext page, it 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 a 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 *explanation variants* method can be implemented by *fragment variants* and *page variants techniques. Page variants* is the most simple adaptive presentation technique. With this technique, a system keeps two or more variants of the same page with different presentations of the same content. As a rule, each variant is prepared for one of possible user stereotypes. When presenting a page, a system select the page variant according to the user stereotype. This technique is used in Anatom-Tutor (Beaumont, 1994) with background stereotypes, in ORIMUHS (Encarnação, 1995b) and WING-MIT(Kim, 1995) with knowledge stereotypes, and in C-book (Kay & Kummerfeld, 1994b) with stereotypes reflecting user proficiency with the language of presentation(English). A similar technique is used in EPIAIM (de Rosis et al., 1993) and C-book to adapt example presentation to the user background. These systems store several examples illustrating particular concept and offers the user the example which is most suitable to the user's previous experience and interests.

Fragment variants is a more fine-grained implementation of *explanation variants* method. An good example here is Anatom-Tutor (Beaumont, 1994). In Anatom-Tutor, a page is not equal to a concept as in some other systems and it can include explanations of several concepts. The system stores several variants of explanations for each concept and the user gets the page which includes variants corresponding to his or her knowledge about the concepts presented in the page. This idea is supported by the work of Paris (Paris, 1988) who shows that users with different knowledge of a particular concept need structurally different explanations about the concept. An interesting feature of Anatom-Tutor is a combination of page variants and fragment variants methods to support adaptation to both the user's background and the user's knowledge. The current page variant for an accessed node is selected according to the user's background. This page can be further adapted: for each concept mentioned in the page, the system selects the explanation which is most suitable to the user's knowledge level.

The most powerful of the all content adaptation techniques is the *frame-based technique* implemented in Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996), and EPIAIM (de Rosis et al., 1993). With this technique all the information about a particular concept is represented in form of a frame. Slots of a frame can contain several explanation variants 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 the 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 and then a subset of slots with high priority is presented in order of decreasing priority. In their conditional parts, these rules can refer not only to the user knowledge level 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. In Hypadapter and EPIAIM, the frame-based technique is used to implement all methods mentioned above except prerequisite and comparative explanation. However, the latter two methods also can be implemented with the frame-based technique by setting appropriate conditions on the knowledge level of related concepts.

The very recent technique developed in the PUSH project (Höök et al., 1996) can be considered as 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 a ordered sequence of typed information entities. Each type of objects in PUSH has its own assortments of information entity types which are used to describe an object of this type. It is very close to the frame-based model where information entities play roles of slots describing various aspects of an object. A specific feature of PUSH, however, is that each information entity is a reasonably big portion of hypertext. The complete description of an object is usually very big and takes several pages of information. To protect users from the information overflow and to help them to find a required piece of information, the system use hiding: it presents only those types of information entities 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 entities 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 can collapse or uncollapse the content of the information entity by clicking on an icon near its title. The resulting interface looks quite 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.

6.3 How adaptive navigation support can help: methods

Adaptive navigation support techniques are used to achieve several adaptation goals: to provide global guidance, to provide local guidance, to support local orientation, to support global orientation, and to help with managing personalized views in information spaces (Table III). Generally, these goals are different, but at the same time each pair of neighboring goals in this list has something in common. So, it is rather a continuum of goals where the borders between neighbors are not clear-cut, and some methods and techniques work for more than one goal (Figure 3).

	Direct guidance	Sorting	Hiding	Annotation	Map adaptation
Global guidance	WebWatcher ITEM/IP ISIS-Tutor SHIVA	Adaptive HyperMan CID HYPERFLEX			
Local guidance	Land Use Tutor HyperTutor	Adaptive HyperMan ELM-PE Hypadapter HYPERFLEX	Hypadapter PUSH	ISIS-Tutor ELM-ART	HYPERCASE
Local orientation support (knowledge)		Hypadapter ELM-PE	[Clibbon] HyperTutor Hypadapter ISIS-Tutor	ELM-ART ISIS-Tutor ITEM/PG Manuel Excel	
Local orientation support (goal)			Hynecosum HyPLAN ISIS-Tutor PUSH SYPROS	ELM-ART ISIS-Tutor	HYPERCASE
Global orientation support			[Clibbon] Hynecosum HyperTutor ISIS-Tutor SYPROS	ITEM/PG ISIS-Tutor ELM-ART Manuel Excel	HYPERCASE

Table III. Adaptive navigation support: goals, technologies, and systems

6.3.1 Global guidance

Global guidance can be provided in hypermedia systems where users have some "global" information goal (i.e., need information which is contained in one or several nodes somewhere in the hyperspace) and browsing is the way to find the required information. The goal of global guidance methods is to help the user to find the shortest way to the information goal with minimal floundering. Global guidance is the primary goal of adaptive navigation support in IR hypermedia and also an important goal in on-line help and on-line information systems with reasonably big hyperspaces. The user's information goal which is usually clearly (Kaplan et al., 1993) or partly (Armstrong et al., 1995; Mathé & Chen, 1996) provided by the user is the primary user feature for adaptive guidance. The most direct method of providing global guidance is to suggest to the user at each step of browsing which of the links from the given node to follow (i.e., to apply the direct guidance technology). This method is used in WebWatcher (Armstrong et al., 1995). A more supportive method is to apply adaptive sorting technology and to sort all the links from the given node according to their relevance to the global goal (the most relevant - first). Here, users still have the possibility of proceeding along the first most relevant link, but also have some more information (relevance of the other links) to make a free choice. This method is used in Adaptive HyperMan (Mathé & Chen, 1996), and HYPERFLEX (Kaplan et al., 1993).

A special case for global guidance is educational hypermedia. In an educational hypermedia system a student usually has a global goal, but it is the learning goal – the knowledge which the user has to learn. The learning goal is different from an information goal, it is not a small piece, but a big subset of the hyperspace (sometimes the total hyperspace). Another specific feature of educational hypermedia is that to provide global guidance a system has to know not only the global learning goal itself, but also the current state of user knowledge of the subject being taught. What is similar with IR hypermedia is the choice of adaptation technologies: it can be either direct guidance or adaptive sorting. A most popular method of providing global guidance in educational hypermedia is direct guidance with the dynamic button "next". There are a number of different elaborated techniques which implement this method. Usually, these techniques are hypermedia-adjusted implementations of curriculum sequencing techniques from the domain of intelligent tutoring systems (ITS). Curriculum sequencing techniques (Brusilovsky, 1992a) can build for a user the shortest individual sequence of learning units (presentations, examples, problems) to achieve the learning goal. The button "next" in educational hypermedia activates a similar sequencing mechanism which selects the node with most relevant educational material according to the current user knowledge, learning goal, and system tutoring strategy (Brusilovsky & Pesin, 1994; Brusilovsky, 1992b; Zeiliger, 1993).

As for the use of adaptive sorting for global guidance in educational hypermedia, at least one method based on sorting links according to the global learning goal was suggested (Tomek et al., 1993), but there are no techniques which implement it. Direct guidance seems to be a very relevant technology for global guidance in educational hypermedia. Users of educational hypermedia are usually novices who have problems with making their own choice and rely heavily on the suggestion of the system. For the novices, the button "next" is the most often used navigation tool (Reed et al., 1995). On the contrary, sorting seems to have a limited applicability in educational hypermedia. First, it can be used only with non-contextual links which are not very popular in educational hypermedia. Second, even for non-contextual links it is not as relevant as in IR hypermedia (where users are mostly professionals) because novices prefer to have a stable order of items (i.e., links) in menus (Debevc et al., 1994; Kaptelinin, 1993).

6.3.2 Local guidance

The goal of local guidance methods is to help the user to make one navigation step by suggesting the most relevant links to follow from the current node. This goal is somewhat similar, but more "modest" than the goal of global guidance. Local guidance methods do not expect a global goal to provide a guidance. They make a suggestion according to the preferences, knowledge, and background of the user – whatever is more important for the given application area. For example, a relevant method of local guidance for IR hypermedia and on-line information systems is sorting links according to user preferences (Adaptive HyperMan, HYPERFLEX) and background (Adaptive HyperMan). Methods used in educational hypermedia are: sorting links according to the user's knowledge (Brusilovsky & Weber, 1996) and direct guidance according to the user knowledge (Kushniruk & Wang, 1994; Pérez et al., 1995). The latter method is usually applied to select the most relevant problem from the set of problems available from the current point. If a system shares the features of both on-line information systems and educational hypermedia like Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996), a local guidance technique can take into account all the features mentioned above: preferences, knowledge, and background.

6.3.3 Local orientation support

The goal of local orientation support methods is to help the user in local orientation (i.e., to help them in understanding what is around and what is his or her relative position in the hyperspace). Existing AH systems implement local orientation support by two different ways: providing additional information about the nodes available from the current node (i.e., use of annotation technology) and limiting the number of navigation opportunities to decrease the cognitive overload and let the users concentrate themselves on analyzing the most relevant links (i.e., use of hiding technology).

Methods based on hiding technology have the same idea: to hide from the user all the links (either from index or from a local node) which are not relevant to him or her at the given moment, or in other words, to show only relevant links. Methods differ with respect to the principles used to decide which links are relevant and which are not. This decision can be made according to the user's knowledge, goals, experience, and preferences. The most simple method in this category is to show only the links relevant to the user's preferences (Waterworth, 1996), but existing implementation of this method is not really adaptive. The most universal method is to show only the links which are relevant to the user's current goal. This method was used in the on-line help system HyPLAN (Grunst, 1993), the institutional hypermedia system Hynecosum (Vassileva, 1996), and in the on-line information systems CID (Boy, 1991) and PUSH (Höök et al., 1996). Another method which is based on the user's experience in the given hyperspace is to show more links for users who have more experience in the hyperspace, as is done in HyperTutor (Pérez, Gutiérrez & Lopistéguy, 1995) and Hynecosum (Vassileva, 1996). Novices in these systems will see quite a small number of links. Along the growth of users' experience they will be able to see gradually more links from the same place.

Two methods based on the hiding technology are specific for educational hypermedia. A very popular method in this application area is to hide the links to the nodes which are not yet ready to be learned (usually it means that these nodes have unlearned prerequisite nodes). This method is used in several systems: ISIS-Tutor (Brusilovsky & Pesin, 1994), HyperTutor (Pérez et al., 1995), (Clibbon, 1995), and Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996). Another method specific for educational hypermedia is to hide the links to the nodes which belong to the educational goals of subsequent lessons and do not belong to the current educational goal, as is done in ISIS-Tutor and SYPROS (Gonschorek & Herzog, 1995).

The idea of methods based on annotation technology is to inform the user about the current "state" of the nodes behind visible links. At present, four basic methods have been suggested for this. First, annotation can be used to show several gradations of link relevancy. For example Hypadapter uses three different font sizes to annotate links as very relevant, relevant, and less relevant. Second, annotation can reflect several levels of user knowledge of the nodes behind the annotated links. At present the techniques implementing this method distinguish three levels of user knowledge of the node: not-known, in-work and well-learned (Brusilovsky & Pesin, 1994; Brusilovsky & Zyryanov, 1993; de La Passardiere & Dufresne, 1992; Schwarz et al., 1996). Two other methods use annotation in situations where hiding is usually used. The first one is outlining the links related to the current goal (Brusilovsky & Pesin, 1994). The second one is providing special annotation for links to not ready to be learned nodes (ITEM/PG and ISIS-Tutor use a kind of dimming and ELM-ART uses "red" traffic light icon). The latter examples show that in many cases methods based on the hiding technology also can be implemented with the annotation technology either by outlining the relevant links or by dimming not relevant links. However, annotation can hardly be used in the cases where the number of visible links is very big (and where hiding is still applicable).

As we see, local orientation support methods do not guide the user directly, but provide help in understanding what are the proximal links and in making well-grounded navigation choices. From another perspective, global and local guidance methods based on sorting technology can support local orientation to some extent, but not as effectively as special orientation support methods described above because the position of a link in a sorted list gives the user very little additional information about this link.

6.3.4 Global orientation support

The goal of global orientation support methods is to help the user to understand the structure of the overall hyperspace and his or her absolute position in it. In non-adaptive hypermedia, this goal is usually achieved by providing visual landmarks and global maps which can directly help the user in global orientation and by providing guided tours to help the user gradually learn the hyperspace (Linard & Zeiliger, 1995). Adaptive hypermedia can provide more support for the user along the same lines by applying hiding and annotation technologies. In fact, all those hiding and annotation methods considered above which are systematic – i.e., when the decision about hiding and annotating a link depends only on the state of the node behind the link and does not depends not from the user's position in the hyperspace – support also user global orientation.

Annotations function as landmark: since a node keeps the same annotation when the user looks at it from different positions in the hyperspace, the user can more easily recognize the nodes he or she met before and understand the current position. Especially useful here is the method which provides different annotations depending on the user knowledge level (Brusilovsky & Pesin, 1994; Brusilovsky & Zyryanov, 1993; de La Passardiere & Dufresne, 1992; Schwarz et al., 1996). Hiding reduces the size of the visible hyperspace and can simplify both orientation and learning. In such application areas as educational or institutional hypermedia where the hyperspace is not especially big, hiding can effectively support gradual learning of the hyperspace. The useful hiding methods for this purpose are those which gradually show the user incrementally larger parts of the hyperspace. The examples from educational hypermedia are the methods of hiding not-ready-to-be-learned nodes as in ISIS-Tutor (Brusilovsky & Pesin, 1994), HyperTutor (Pérez et al., 1995), (Clibbon, 1995), and Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996) and nodes which are the educational goals of subsequent lessons as in ISIS-Tutor and SYPROS (Gonschorek & Herzog, 1995). An example of a more universal method is the method which shows gradually more links with the growth of the user experience in the given hyperspace, as is done in HyperTutor and Hynecosum (Vassileva, 1996).

A promising direction of adaptive global orientation support is *adaptation of local and global maps* (Zyryanov, 1996), but this direction is really underinvestigated. Generally, all systematic hiding and annotation methods can be applied to adapt hyperspace maps, but no existing AH systems really do that. Another possibility is applying the map adaptation technology (i.e., the adaptive construction of local and global maps where the very structure of the map can depend on the user characteristics). The only existing work in this area is the HYPERCASE system (Micarelli & Sciarrone, 1996) which suggests a technique to generate global and local maps according to the current user goal.

6.3.5 Managing personalized views

Managing personalized views is a new goal for adaptive hypermedia systems. Personalized views is a way to organize an electronic workplace for the users who need a access to a reasonably small part of a hyperspace for their everyday work. The usual way to protect these users from the complexity of the overall hyperspace is to let them organize personalized goal-oriented views. Each view is just a list of links to all hyperdocuments which are relevant to a particular working goal. Traditionally, it is the duty of the users to create and to manage their personalized views (adaptability). Classic hypermedia systems and modern WWW browsers suggest bookmarks and hotlists as a way to make personal views. More advanced systems suggest some more high-level adaptability mechanisms based on metaphors (Waterworth, 1996) and user models (Vassileva, 1996).

Adaptive solutions, i.e., system supported management of personalized views, are required in WWW-like dynamic information spaces where items can appear, disappear, or evolve. By now we can mention only BASAR (Thomas & Fischer, 1996) as an example of a system which provides adaptive management of personalized views. BASAR uses intelligent agents to collect and maintain an actual set of links relevant to one of the user's goals. The agents can search regularly for new relevant items and identify expired or changed items. Currently we can hardly formulate any methods (as generalization of existing techniques) for managing personalized views, but it can be done very soon because several research groups are start working on BASAR-like systems for WWW.

6.4 How adaptive navigation support can help: techniques

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

A powerful and universal technique for ordering the links according their relevance is implemented in Adaptive HyperMan (Mathé & Chen, 1996). The system takes into account many factors or *inputs*: user background (profile), user search goal (set of keywords), current node of interest, etc., and returns as an output an ordered set of documents relevant to the provided input. If the search goal is set, the system can provide global guidance, otherwise it can provide local guidance. To calculate the relevancy for the documents the system uses a personal relevance network which stores in this special form individual preferences of the users. To provide feedback, the user judges the set of nodes selected by the system as relevant or not relevant to the input from his or her personal point of view. This feedback is used by the system to update the individual relevance network so it gradually learns user preferences.

A flexible original (and actually the first known) technique for local guidance and local orientation support is implemented in Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996). A node in Hypadapter is represented as a frame which usually has several "relation" slots storing different kinds of links to related nodes. Hypadapter uses a set of rules to calculate the relevance of links for each relational slot from the current node. Each applicable rule can increment or decrement the relevance value of the link depending on specific features of the link and the node behind it. Then the system uses a combination of sorting, hiding, and annotation to show relevance to the user: first, the links are sorted according to the relevance, second, the system uses three different font sizes to annotate very relevant, relevant, and less relevant links, third, the system hides irrelevant links. For example, the links to concepts not ready to be learned can be hidden in this way.

Another example of a rule-based adaptive navigation support technique is the rule-based hiding technique implemented in the educational hypermedia systems HyperTutor (Pérez et al., 1995) and

SYPROS (Gonschorek & Herzog, 1995). Both systems use special sets of pedagogical rules to decide which concepts and nodes should be visible at a given moment and which should not. These rules take into account the type of the concept, the types of its links to other concepts, and the current state of the user's knowledge as reflected in the user model. If a node is not to be visible, then all contextual links to this node will be changed from hot words to normal text. The rule-based hiding technique is very flexible. By manipulating with hiding rules it is possible to implement several local and global orientation support methods based on hiding, such as hiding not ready to be learned nodes, or hiding nodes which belong to the subsequent lessons. In SYPROS hiding is used to implement the *explanation variants* method of content adaptation. The hyperspace contains several nodes devoted to the same concepts which are oriented toward users with different levels of knowledge. Hiding rules always keep visible only those nodes which suit to the user's current level of knowledge.

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 higher 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. Working with Hynecosum the user always informs the system which task he or she is currently performing by selecting one of the tasks from a personal hierarchy. Knowing the user's task, the system can hide from the user all nodes which are not relevant to the current task. This makes the set of visible nodes manageable. When the selected task is a high level task in the hierarchy, the number of visible nodes can still be too big. To provide additional support for novices, Hynecosum will not show links to document nodes until the user sets the current task more exactly by selecting a lower level task in the hierarchy. The more experienced the user is, the higher the level is when the user gets access to document nodes. Thus, the technique implemented in Hynecosum supports both goal-based and experience-based methods of hiding.

The techniques implemented in ISIS-Tutor (Brusilovsky & Pesin, 1994), ITEM/PG (Brusilovsky & Zyryanov, 1993), and ELM-ART (Schwarz, Brusilovsky & Weber, 1996) support several methods of local and global orientation support based on annotation and hiding. 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 in the conceptual network is represented by a node of the hyperspace. Concept nodes in the network are connected by different kinds of relationships such as "is-a", "part-of", and "prerequisite". Another kind of node in the hyperspace is a learning unit (such as a textbook section, a problem, or an example). Each learning unit is connected to all concept nodes which are required to work with this unit. The overlay student model separately represents 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 would 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 (ITEM/PG and ISIS-Tutor) and different icons (ELM-ART). As an option, the links to not-ready-to-be-learned pages in ISIS-Tutor can be hidden.

The same framework was used in ISIS-Tutor and ELM-ART to implement one additional adaptation method based on the learning goal. The teacher can set for each student a sequence of learning goals (here a goal is a set of concept nodes of the network which have to be learned). Each goal serves as a chapter in a book: Concepts belonging 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. Two technologies are used for adaptation to the learning goal: first, to attract the student's attention the system outlines links to the concepts belonging to the current goal (ISIS-

Tutor, ELM-ART), second, to decrease the student's cognitive load it hides concepts which belong to the next learning goals (ISIS-Tutor).

The technique used in HYPERCASE (Micarelli & Sciarrone, 1996) is the only known example of map adaptation. This technique supports local and global orientation by adapting the form of local and global maps to the didactic or information goal of the users. Similar to HYPERFLEX, HYPERCASE represents and uses knowledge about possible goals for goal adaptation, but the type of goal knowledge is very different from HYPERFLEX. HYPERCASE uses a case-based approach and a neural network technology to store in the database of cases several typical navigation paths for each of the didactic goals. Using this knowledge, the system can find the most similar standard path (and thus the most probable didactic goal) for the navigation path of a real student supplied as an input to the case-based mechanism. When the student requests help, HYPERCASE can show where he or she is located in the hyperspace by drawing a wide-area or local area hierarchical map using the technique suggested in (Rivlin et al., 1994). As the root of the hierarchy the system uses the "central node" of the hyperspace (which is computed by a special method) for the wide-area map and the closest node of a deduced standard path for the local-area map. This technique is suitable for educational hypermedia and on-line information systems.

7 User modeling in adaptive hypermedia.

All previous sections were devoted to the issues related to adaptation techniques in adaptive hypermedia (i.e., techniques which use the student model to provide some kind of adaptation to the user). This is, however, only the second part of the overall adaptation process (Figure 1). In this chapter we will consider some issues related with the first part of this process - the user modeling. We will not discuss here all issues of user modeling in adaptive hypermedia because most of them are not really specific to adaptive hypermedia (unlike the issues related with adaptation) and have been discussed in a number of other papers on adaptive systems - see (Kobsa, 1993; Kok, 1991) for a comprehensive review. What seems to be quite specific for user modeling in adaptive hypermedia is the sharing of the duties between the user and the system in the process of user modeling. The discussion in this section is centered around the problems of collaborative user modeling in AH systems.

A recent review of adaptive user interfaces (Dieterich et al., 1993) provides a very fine-grained taxonomy to describe sharing duties between the system and the user in the process of adaptation. We will base our discussion on a different model which is more suitable for our domain. We outline three stages in the adaptation process (Figure 1): collecting data about the user, processing the data to build or update the user model, and applying the user model to provide the adaptation. In our case, the last stage of this process is always performed by the system because we define as adaptive hypermedia systems exactly the systems which can provide automatic adaptation on the basis of the user model. In the case of "classic" or fully adaptive systems, the system also performs both of the previous stages of the overall process: while the user is simply working in an application system, the adaptation component watches what the user is doing, collects the data describing user's activity, process these data to build the user model, and then provides an adaptation. Unfortunately, such an ideal situation is very rarely met in adaptive hypermedia systems. Almost all of them rely on external sources of information about the user. The main source of information for many of the systems is the information provided by the users themselves in several forms.

7.1 Problems with automatic user modeling in hypermedia systems

There are some general problems related to automatic user modeling in adaptive systems. We name only two of them. First, automatic user modeling is not completely reliable. As it was pointed out in (de Rosis et al., 1993), systems which perform both user modeling and adaptation without user influence are twice unreliable - they can make an error when deducing the student model and they can make an error when providing adaptation (even if the model itself is correct). Second, some components of the user model, such as background and preferences of the user, can not be deduced at all and have to be provided directly by the user. At the same time, research on adaptive dialogue systems demonstrates a number of effective technologies of automatic user modeling. In particular, two widely used standard technologies which have been shown to be effective for different applications involve (1) tracking the user's actions to understand which commands and concepts are known to the user and which are not, and (2) using the plan knowledge base to deduce the goal of the user.

The situation with automatic user modeling in adaptive hypermedia seems to be specific. Unlike in other kinds of application systems, watching what the user is doing in hypermedia provides insufficient information for user modeling. This fact was emphasized in several works on adaptive hypermedia (Beaumont, 1994; Boyle & Encarnacion, 1994; Vassileva, 1996). Most known techniques for automatic user modeling can hardly be applied in this area. The only information about the user's actions which the system can record is the user's path through the hyperspace and the time spent on each node. The user's path itself and patterns of user navigation are an interesting source of information, but it is very hard to update the user model using only this information. There are only two adaptive hypermedia systems which offer some techniques for this. Hynecosum can deduce the user level of experience based on the patterns of user navigation and HYPERCASE (Micarelli & Sciarrone, 1996) can deduce the user's didactic goal from the user's path. Some cumulative information about user navigation, such as the time spent on a particular node or the number of visits to it is easier to use. For example, HYPERFLEX uses the time spent on a node to measure how relevant the node is to the current goal, and ISIS-Tutor uses the number of visits to predict the user knowledge level of the visited concept nodes. However, the fact that the user has visited a page several times or spent reasonable time on it does not even guarantee that the user attentively read its content. This kind of information is not reliable and can not be used as the only source for building the user model. In fact, both HYPERFLEX and ISIS-Tutor use this information along with additional sources to update the user model.

7.2 Additional sources of information for automatic user modeling

If automatic user modeling in adaptive hypermedia is hard and unreliable what are the ways for these systems to learn the information about the user? The answer depends on the application area. Two kinds of adaptive hypermedia systems - on-line help systems and educational hypermedia in intelligent tutoring systems - are more lucky to have an additional source of information to update the user model (Figure 5). On-line help systems are designed to serve as a part of a complex interactive system. Regular methods of user modeling can be applied in these systems to recognize the goal and the level of user experience. A good example is provided by HyPLAN (Grunst, 1993) which can infer the goal of the user working in a spreadsheet application and offer an adaptive help index which shows only the links relevant to this goal. Educational hypermedia in intelligent tutoring systems can use the information about the user (i.e., the student) from the student model which is an important part of any ITS. The student model in ITS is updated mainly by analyzing the student's answers to the offered tests, questions, or problems. This information about the student is usually more reliable than information inferred by watching the student navigation. It is not surprising that most existing adaptive educational hypermedia systems (Anatom-Tutor, ELM-ART, ELM-PE, HyperTutor, ISIS-Tutor, ITEM/PG, Land Use Tutor, SHIVA, SYPROS) are designed in ITS context.

7.3 Three approaches to collaborative user modeling

In all other cases, the only way for the system to get the required information about the user is to involve the user in the process of user modeling and to collaborate with the user in gathering the information. It can be called *collaborative* or *cooperative* (Kay, 1995) user modeling. According to the three stages of user modeling process, we distinguish three approaches to involving the user in the process of user modeling (Figure 5). First, users can provide the data required for the user modeling mechanism. For example, rather than guessing whether a particular page is relevant to the user's goal using the time spent on it (as it was done in HYPERFLEX) the system can directly request this feedback from the user (as it was done in Adaptive HyperMan). Similarly, rather than guessing whether the user understands a particular page, the system can get feedback from the user. Such a feedback is a more reliable source of information and not very difficuilt for the user to provide. Data received from the user can be further processed by the system to update the user model.

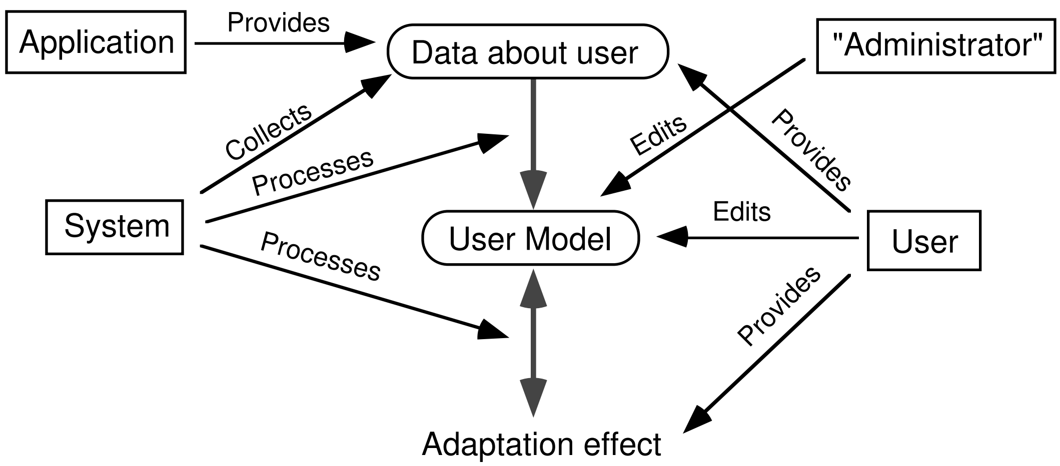

Figure 5. Collaborative user modeling in adaptive hypermedia

Second, users can make the desired adaptation themselves, directly showing the system what they would like to see on the screen in the given context. HYPERFLEX (Kaplan et al., 1993), MetaDoc (Boyle & Encarnacion, 1994), KN-AHS (Kobsa et al., 1994), and PUSH (Höök et al., 1996) demonstrate how this can be done in adaptive hypermedia for both link and content adaptation. HYPERFLEX adapts the list of links to related nodes by sorting the links according to their relevance to the current node and current user goal. The users can drag the links in the list to change their order thus telling the system which order of links they expect to see for the current goal and node. HYPERFLEX takes this information into account updating the main part of the user model – the relevance matrix. MetaDoc, KN-AHS, and PUSH adapt the context of a stretchtext page before presenting it to the user by showing or hiding some pieces of stretchtext according to the user's knowledge. Users can further adapt the content of the page by collapsing or uncollapsing pieces of stretchtext thus showing the system which information they actually want (or do not want) to see on the given page. These changes let MetaDoc and KN-AHS update the user knowledge level in the user model and help PUSH to infer the user's goal.

To proceed according to this second approach the system has to provide users with special interface features to make the adaptation (such as the possibility of dragging the links or collapsing/uncollapsing the text), i.e., the system should be user-adaptable. At the same time, the system has to be able to update the user model according to the preferences demonstrated by the user-driven adaptation. As a result, the preferences demonstrated by the user in one context can be used to adapt the interface in other contexts. The latter feature distinguish systems like HYPERFLEX and MetaDoc from simple user-adaptable systems which just store the set of user preferences "as is" and which can not apply it in different contexts.

Third, the information in the user model can be updated directly by the information received from the users. In adaptive hypermedia systems Adaptive HyperMan (Mathé & Chen, 1996), Hypadapter (Hohl, Böcker & Gunzenhäuser, 1996), Anatom-Tutor (Beaumont, 1994), EPIAIM (de Rosis et al., 1993), and C-Book (Kay & Kummerfeld, 1994b) as well as in many other kinds of adaptive systems, this method is used to set various features related to user background (profession, relevant experience, prospects, etc.) which can be provided only by the users themselves. More specific to adaptive hypermedia systems (Armstrong et al., 1995; Höök et al., 1996; Kaplan et al., 1993; Mathé & Chen, 1996; Thomas & Fischer, 1996; Vassileva, 1996) is the possibility for users to set their current goals (which are hard for the system to infer). The final step here is to make the user model viewable and changeable (Cook & Kay, 1994; Kay, 1995) – i.e., to provide a special interface for the users to inspect the content of the user model and to update it. With such an interface, the users can inspect all aspects of their models (Hohl, Böcker & Gunzenhäuser, 1996) and make quite complicated changes, for example, changing the personal task hierarchy and visibility of nodes (Vassileva, 1996). In some cases, end users are not skilled enough to change a nontrivial part of the user model reliably while this part is very important for the correct functioning of the adaptation. In these cases the system can provide an interface for a skilled responsible person (administrator) to set the model correctly (Figure 5). For example, setting user classes and hierarchies in Hynecosum is expected to be done by an administrator. In educational hypermedia systems, the role of such an administrator is usually played by the teacher. In ISIS-Tutor it is the role of the teacher to set individual sequences of learning goals (an important part of the student model) for each student.

What is similar for the three analyzed approaches is the idea of involving the user in the process of user modeling to get additional information from the user and - as a result - to make user modeling more simple and more reliable. What is different with these approaches is the applicability. Providing feedback is the most simple method for users. It requires very small changes in the interface and can be used with any category of users. This method can provide some reliable data for inferring student preferences or knowledge. From another perspective, the amount of information provided with the feedback method is limited. Providing feedback is a side activity for users and should not disturb them from their main work. Feedback which requires more than two mouse clicks can be considered obtrusive. However, even one-click feedback combined with a powerful user modeling technique can give good results as demonstrated by Adaptive HyperMan (Mathé & Chen, 1996).

Receiving information for the user model directly from the student is the most simple method for the system. It requires a rather complicated interface but does not require complicated inference methods. It is definitely the most relevant method of getting the information which can be provided easily - such as goal or background aspects. Updating some more complicated data as an overlay knowledge model or a goal hierarchy requires considerable time and experience level. Understanding the relationships between updated information in the user model and changed behavior also implies reasonable experience. This method is suitable for an experienced user who uses the system in everyday work or for an administrator.

The method based on user-driven adaptation is in the middle between the above extremes. To provide adaptation requires more time for the user than to provide feedback, but it is still not very distracting and can be done by a not very skilled user (because the user has to provide just a desired effect, not the changes in the user model). Data provided by the user still have to be processed to update the student model, but the amount of information provided is greater and the processing methods can not be as complicated. This way is also the most transparent one for users because the relation between their actions and the received adaptation is straitforward. Finally, it naturally joins adaptivity and adaptability (Fischer, 1993; Höök et al., 1996) and opens the way to apply such a promising method as adaptively supported adaptability (Oppermann, 1994a; Oppermann, 1994b). We think that this approach is the most relevant one for the continuum of users excluding beginners on one side (for them feedback is more easy) and experienced professionals on the other side (they can get more control by updating the user model).

8 Concluding remarks

Adaptive hypermedia is a new but very quickly developing area of research. More than 20 truly adaptive hypermedia systems have been developed and described within the last 3 years. By now, most existing AH systems are applied in the traditional hypertext and hypermedia areas, such as on-line information systems, on-line help systems, and educational hypermedia. Some recent work demonstrates that AH techniques can be applied in a number of other application areas: IR hypermedia, institutional hypermedia, and personalized information spaces. In each of these application areas AH techniques can help to solve several groups of problems related to navigation in large hyperspaces and differences between hypermedia users.

Though adaptive hypermedia is a new area where most existing AH systems are original and very different from each other, it is already possible to identify a set of effective AH methods which have been implemented in several systems. These methods can be used as a set of building tools for creating future AH systems. In real AH systems, these methods have been implemented by different techniques which use different knowledge representations and different adaptation algorithms. Many of these techniques are well-described in the literature and can be re-used by the designers of AH systems.

The main characteristic of all methods and techniques of adaptive hypermedia is the applied technology of adaptation. We have identified seven adaptation technologies which are different from a "what can be adapted" point of view: adaptive text presentation, direct guidance, adaptive sorting, hiding, and annotation of links, adaptive multimedia presentation, and map adaptation. The first five of these technologies are well-investigated and used by many AH methods and techniques. The last two technologies are currently under-investigated and need further research. More work has to be done in applying current techniques of adaptive multimedia presentation to real-world hyperspaces and in making adaptive known techniques for generation of hypermedia maps.

One of the special features of AH systems is the important role of users in the user modeling process. Most existing AH systems can not rely on "classic" automatic user modeling and must use external sources of information about the user. Many of the systems extend traditional "look and browse" hypermedia interface with special features that let users provide more information about their goals, knowledge, and preferences. A promising approach here is to let the user adapt presented hypermedia pages and take the user's changes into account to update the user model.

This paper is an attempt at conducting a comprehensive review of existing AH systems. The review classifies existing AH techniques and methods according to several introduced dimensions and systematically presents the most interesting AH methods and techniques for both the adaptation and user modeling parts of the overall adaptation loop. At the end, we want to mention three important issues related with the topic of the paper which have been left outside the scope of the review. These issues are currently underinvestigated. For each of them we can name 2-4 papers which are useful as a background for further research but can not be used for providing a comprehensive review. We consider the three issues listed below as the important directions of research in AH area.

The first issue is the efficiency of AH methods and techniques. Unfortunately, the situation with experimental testing of AH systems is not as advanced as the situation with development of these systems. As we have shown, a good number of interesting techniques were suggested and implemented in AH systems. The problem is that very few of them have been evaluated by a properly designed experiment. By now, we can name only three systems which have been tested by a comprehensive experiment with the number of subjects large enough to get statistically significant data: MetaDoc (Boyle & Encarnacion, 1994), HYPERFLEX (Kaplan et al., 1993), and ISIS-Tutor (Brusilovsky & Pesin, 1995). The lack of experimental studies is a weak point of AH research and more work has to be done in this direction.

The second issue which is related to the previous one is the usability of adaptation. This issue is important for any adaptive computer system. Adaptivity is not a silver bullet. Adding adaptivity does not always makes a system better, moreover, it can make system much less usable if the users will not be able to understand what is going on in an adaptive system and lose the control over the system. What are the conditions and features of "usable" adaptation? Currently, very few papers contribute directly to this issue. We should specially mention the paper (Vassileva, 1996) which introduces the idea of "stepwise" adaptation and the papers (Höök et al., 1996; Kay, 1995) which discuss the problems of "transparent" adaptation.

The third issue to mention is the relations between AH research and the World Wide Web. WWW, a hyperspace spread over the Internet, has potentially unlimited number of nodes and variety of users. WWW is currently a promising direction for application of AH techniques. Moreover, we think that WWW will be the most important factor stimulating AH research in the next several years. In fact, several AH systems reviewed in this paper are completely or partly implemented on WWW (Armstrong et al., 1995; Höök et al., 1996; Schwarz et al., 1996; Thomas & Fischer, 1996). These works are very different and can hardly be used to provide some generalizations, for example, which adaptation techniques are most suitable for WWW. However, they already show some useful and re-usable ways of implementing AH methods and techniques on WWW.

Acknowledgments

I would like to thank Julita Vassileva, Gerhard Weber, Kristina Höök, Richard Keller and John Eklund for providing comments on the earlier version of this paper. Special thanks to Alfred Kobsa for his permanent support in preparing this review. I would like to thank also the reviewers of this paper for their useful and constructive comments. Part of this work was supported by an Alexander von Humboldt-Stiftung Fellowship to the author.

Appendix 1: List of adaptive hypermedia systems

"Real" AHS

Adaptive HyperMan (Mathé & Chen, 1994, Mathé & Chen, 1996),
Anatom-Tutor (Beaumont, 1994),
Basar (Thomas, 1995; Thomas & Fischer, 1996),
C-Book (Kay & Kummerfeld, 1994a; Kay & Kummerfeld, 1994b),
CID (Boy, 1991),
[Clibbon] (Clibbon, 1995),
DHS (Shibata & Katsumoto, 1993; Katsumoto et al., 1994; Katsumoto et al., 1996),
ELM-ART (Brusilovsky, Schwarz & Weber, 1996; Schwarz, Brusilovsky & Weber, 1996),
ELM-PE (Brusilovsky & Weber, 1996),
EPIAIM (de Rosis et al., 1993; de Rosis, De Carolis & Pizzutilo, 1994),
Hynecosum (Vassileva, 1994, Vassileva, 1996),
Hypadapter (Böcker et al., 1990; Hohl, Böcker & Gunzenhäuser, 1996),
HYPERCASE (Micarelli & Sciarrone, 1996),
HYPERFLEX (Kaplan et al., 1993),
HyperTutor (Pérez et al., 1995; Pérez et al., 1995),
HyPLAN (Fox, Grunst & Quast, 1993; Grunst, 1993),
ISIS-Tutor (Brusilovsky & Pesin, 1994; Brusilovsky & Pesin, 1995),
ITEM/PG (Brusilovsky, Pesin & Zyryanov, 1993; Brusilovsky & Zyryanov, 1993),
KN-AHS (Kobsa et al., 1994),
Land Use Tutor (Kushniruk & Wang, 1994),
Lisp-Critic (Fischer et al., 1990),
Manuel Excel (de La Passardiere & Dufresne, 1992)
MetaDoc (Boyle & Encarnacion, 1994),
ORIMUHS (Encarnação, 1995a; Encarnação, 1995b),
PUSH (Höök et al., 1996),
SYPROS (Gonschorek & Herzog, 1995)
SHIVA (Zeiliger, 1993),
WebWatcher (Armstrong et al., 1995),
WING-MIT (Kim, 1995).

Some systems close to AHS

ITEM/IP (Brusilovsky et al., 1993; Brusilovsky, 1992b)
Information Islands (Waterworth, 1996)
PPP (André & Rist, 1996)
[Yetim] (Yetim, 1993)

References

Agosti, M., M. Melucci, and F. Crestani: 1995, 'Automatic authoring and construction of hypermedia for information retrieval'. *Multimedia Systems* **3** (1), 15-24.

André, E. and T. Rist: 1996, 'Towards a New Generation of Hypermedia Systems: Extending Automated Presentation Design for Hypermedia', Report, DFKI GmbH, Saarbrücken.

Arens, S. and R. Hammwöhner: 1995, 'Ein graphischer Browser für das Konstanzer Hypertext-System'. In: R. Kuhlen and M. Ritterberg (eds.): *Hypertext - Information Retrieval - Multimedia*. Konstanz: Universitätsverlag Konstanz, pp. 175-189.

Armstrong, R., D. Freitag, T. Joachims, and T. Mitchell: 1995, 'WebWatcher: A learning apprentice for the World Wide Web'. *AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments*, Stanford, CA, http://www.isi.edu/sims/knoblock/sss95/mitchell.ps.

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

Benyon, D. R. and D. M. Murray: 1993, 'Applying user modeling to human-computer interaction design'. *AI Review* **6**, 43 - 69.

Böcker, H.-D., H. Hohl, and T. Schwab: 1990, 'ϒπAdaptερ - Individualizing Hypertext'. In: D. Diaper (ed.) *INTERACT'90.* Amsterdam: North-Holland, pp. 931-936.

Boy, G. A.: 1991, 'On-line user model acquisition in hypertext documentation'. *IJCAI'91 Workshop 'Agent Modeling for Intelligent Interaction'*, Sydney, Australia, pp. 34-42.

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

Brusilovsky, P. and L. Pesin: 1994, 'ISIS-Tutor: An adaptive hypertext learning environment'. *JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering*, Pereslavl-Zalesski, Russia, pp. 83-87.

Brusilovsky, P. and L. Pesin: 1995, 'Visual annotation of links in adaptive hypermedia'. *CHI'95 Conference Companion*, Denver, pp. 222-223.

Brusilovsky, P., L. Pesin, and M. Zyryanov: 1993, 'Towards an adaptive hypermedia component for an intelligent learning environment'. In: L. J. Bass, J. Gornostaev and C. Unger (eds.): *Human-Computer Interaction*. Lecture Notes in Computer Science, Vol. 753, Berlin: Springer-Verlag, pp. 348-358.

Brusilovsky, P., E. Schwarz, and G. Weber: 1996, 'ELM-ART: An intelligent tutoring system on World Wide Web'. *Third International Conference on Intelligent Tutoring Systems, ITS-96*, Montreal.

Brusilovsky, P. and G. Weber: 1996, 'Collaborative example selection in an intelligent example-based programming environment'. *International Conference on Learning Sciences, ICLS-96*, Evanston, USA.

Brusilovsky, P. and M. Zyryanov: 1993, 'Intelligent tutor, environment and manual for physical geography'. *Seventh International PEG Conference*, Edinburgh, pp. 63-73.

Brusilovsky, P. L.: 1992a, 'A framework for intelligent knowledge sequencing and task sequencing'. In: C. Frasson, G. Gauthier and G. I. McCalla (eds.): *Intelligent Tutoring Systems*. Berlin: Springer-Verlag, pp. 499-506.

Brusilovsky, P. L.: 1992b, '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'. *CHI'95 Conference Companion*, Denver, pp. 224-225.

Cook, R. and J. Kay: 1994, 'The justified user model: a viewable, explained user model'. *4-th International Conference on User Modeling*, Hyannis, MA, pp. 145-150.

de La Passardiere, B. and A. Dufresne: 1992, 'Adaptive navigational tools for educational hypermedia'. In: I. Tomek (ed.) *Computer Assisted Learning*. Berlin: Springer-Verlag, pp. 555-567.

de Rosis, F., B. De Carolis, and S. Pizzutilo: 1993, 'User tailored hypermedia explanations'. *INTERCHI'93 Adjunct proceedings*, Amsterdam, pp. 169-170.

de Rosis, F., B. De Carolis, and S. Pizzutilo: 1994, 'User tailored hypermedia explanations'. *Workshop Adaptive Hypertext and Hypermedia at 4-th International Conference on User Modeling*, Hyannis, MA, http://www.cs.bgsu.edu/hypertext/adaptive/deRosis.html.

Debevc, M., S. Rajko, and D. Donlagic: 1994, 'Adaptive bar implementation and ergonomics'. *Informatica : Journal of Computing and Informatics* **18**, 357-366.

Dieterich, H., U. Malinowski, T. Kühme, and M. Schneider-Hufschmidt: 1993, 'State of the art in adaptive user interfaces'. In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (eds.): *Adaptive user interfaces: Principles and practice*. Amsterdam: North-Holland, pp. 13-48.

Encarnação, L. M.: 1995a, 'Adaptive help for interactive graphics systems: an application-independent approach'. *Workshop Adaptivität und Benutzermodellierung in interactiven Systemen (ABIS 95)*, München.

Encarnação, L. M.: 1995b, 'Adaptivity in graphical user interfaces: An experimental framework'. *Computers & Graphics* **19** (6), 873-884.

Fischer, G.: 1993, 'Shared knowledge in cooperative problem-solving systems - integrating adaptive and adaptable components'. In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (eds.): *Adaptive user interfaces: Principles and practice*. Amsterdam: North-Holland, pp. 49-68.

Fischer, G., T. Mastaglio, B. Reeves, and J. Rieman: 1990, 'Minimalist explanations in knowledge-based systems'. *23-th Annual Hawaii International Conference on System Sciences*, Kailua-Kona, HI, pp. 309-317.

Fox, T., G. Grunst, and K.-J. Quast: 1993, 'HyPlan - a context-sensitive hypermedia help system', Report No. 743: *Arbeitspapiere der GMD*, GMD, Germany.

Furnas, G. W.: 1986, 'Generalized fisheye views'. *CHI'86*, pp. 16-23.

Gonschorek, M. and C. Herzog: 1995, 'Using hypertext for an adaptive helpsystem in an intelligent tutoring system'. *AI-ED'95, 7th World Conference on Artificial Intelligence in Education*, Washington, DC, pp. 274-281.

Greer, J. and G. McCalla (eds.): 1993, 'Student modeling: the key to individualized knowledge-based instruction'. NATO ASI Series F, Vol. 125, Berlin: Springer-Verlag.

Grunst, G.: 1993, 'Adaptive hypermedia for support systems'. In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (eds.): *Adaptive user interfaces: Principles and practice*. Amsterdam: North-Holland, pp. 269-283.

Hammond, N.: 1989, 'Hypermedia and learning: Who guides whom?'. In: H. Maurer (ed.) *Computer Assisted Learning*. Lecture Notes in Computer Science, Vol. 360, Berlin: Springer-Verlag, pp. 167-181.

Helmes, L., M. Razum, and A. Barth: 1995, ‘Concept of a hypertext interface for the information retrieval in complex factual databases’. In: R. Kuhlen and M. Ritterberg (eds.): *Hypertext - Information Retrieval - Multimedia*. Konstanz: Universitätsverlag Konstanz, pp. 175-189.

Hohl, H., H.-D. Böcker, and R. Gunzenhäuser: 1996, ‘Hypadapter: An adaptive hypertext system for exploratory learning and programming’. *User Models and User Adapted Interaction* **6** (this issue).

Höök, K., J. Karlgren, A. Wærn, N. Dahlbäck, C. G. Jansson, K. Karlgren, and B. Lemaire: 1996, ‘A glass box approach to adaptive hypermedia’. *User Models and User Adapted Interaction* **6** (this issue).

Kaplan, C., J. Fenwick, and J. Chen: 1993, ‘Adaptive hypertext navigation based on user goals and context’. *User Models and User Adapted Interaction* **3** (3), 193-220.

Kaptelinin, V.: 1993, ‘Item recognition in menu selection: The effect of practice’. *INTERCHI'93 Adjunct Proceedings*, Amsterdam, pp. 183-184.

Katsumoto, M., M. Fukuda, N. Irie, and Y. Shibata: 1994, ‘Dynamic hypermedia system based on Perceptional Link Method for distributed design image database’. *9th International Conference on Information Networking (ICOIN-9)*, pp. 49-54.

Katsumoto, M., M. Fukuda, and Y. Shibata: 1996, ‘The Kansei Link Method for Multimedia Database’. *10th International Conference on Information Networking (ICOIN-10)*, pp. 382-389.

Kay, J.: 1995, ‘The UM toolkit for cooperative user models’. *User Models and User Adapted Interaction* **4** (3), 149-196.

Kay, J. and R. Kummerfeld: 1994a, ‘Adaptive hypertext for individualised instruction’. *Workshop on Adaptive Hypertext and Hypermedia at Fourth International Conference on User Modeling*, Hyannis, MA, http://www.cs.bgsu.edu/hypertext/adaptive/Kay.html.

Kay, J. and R. J. Kummerfeld: 1994b, ‘An Individualised Course for the C Programming Language’. *Second International WWW Conference "Mosaic and the Web"*, Chicago, IL, http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.html.

Kim, D.-W.: 1995, ‘WING-MIT: Das auf einer multimedialen und intelligented Benutzerschnittstelle basierende tutorielle Hilfesystem für das Werkstoffinformationssystem WING-M2’. *Workshop Adaptivität und Benutzermodellierung in interactiven Systemen (ABIS 95)*, München.

Kobsa, A.: 1993, ‘User modeling: Recent work, prospects and hazards’. In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (eds.): *Adaptive user interfaces: Principles and practice*. Amsterdam: North-Holland, pp. 111-128.

Kobsa, A., D. Müller, and A. Nill: 1994, ‘KN-AHS: An adaptive hypertext klient of the user modeling system BGP-MS’. *4-th International Conference on User Modeling*, Hyannis, MA, pp. 31-36.

Kok, A. J.: 1991, ‘A review and synthesis of user modeling in intelligent systems’. *The Knowledge Engeneering Review* **6** (1), 21-47.

Kushniruk, A. and H. Wang: 1994, ‘A hypermedia-based educational system with knowledge-based guidance’. *ED-MEDIA'94 - World conference on educational multimedia and hypermedia*, Vancouver, Canada, pp. 335-340.

Linard, M. and R. Zeiliger: 1995, ‘Designing a navigational support for educational software’. In: B. Blumental, J. Gornostaev and C. Unger (eds.): *Human-Computer Interaction*. Lecture Notes in Computer Science, Vol. 1015, Berlin: Springer-Verlag, pp. 63-78.

Mathé, N. and J. Chen: 1994, ‘A user-centered approach to adaptive hypertext based on an information relevance model’. *4-th International Conference on User Modeling*, Hyannis, MA, pp. 107-114.

Mathé, N. and J. Chen: 1996, 'User-centered indexing for adaptive information access'. *User Models and User Adapted Interaction* **6** (this issue).

Maybury, M. T. (ed.) 1993, 'Intelligent Multimedia Interfaces'. Boston: AAAI Press/MIT Press.

Micarelli, A. and F. Sciarrone: 1996, 'A case-based toolbox for guided hypermedia navigation'. *Fifth International Conference on User Modeling, UM-96*, Kailua-Kona, Hawaii, pp. 129-136.

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

Mukherjea, S. and J. D. Foley: 1995, 'Visualizing the World-Wide Web with navigational view builder'. *Third International World-Wide Web Conference. In: Computer Networks and ISDN Systems* **27** (6), 1075-1087.

Mukherjea, S., J. D. Foley, and S. Hudson: 1995, 'Visualizing complex hypermedia networks through multiple hierarchical views'. *CHI'95*, Denver, pp. 331-337.

Oppermann, R. (ed.) 1994a, 'Adaptive user support - ergonomic design of manually and automatically adaptable software'. Computers, Cognition, and Work, Vol. 40, Hillsdale, NJ: Lawrence Erlbaum Associates.

Oppermann, R.: 1994b, 'Adaptively supported adaptability'. *International Journal on Human-Computer Studies* **40**, 455-472.

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

Pérez, T., J. Gutiérrez, and P. Lopistéguy: 1995, 'An adaptive hypermedia system'. *AI-ED'95, 7th World Conference on Artificial Intelligence in Education*, Washington, DC, pp. 351-358.

Pérez, T., P. Lopistéguy, J. Gutiérrez, and I. Usandizaga: 1995, 'HyperTutor: From hypermedia to intelligent adaptive hypermedia'. *ED-MEDIA'95, World conference on educational multimedia and hypermedia*, Graz, Austria, pp. 529-534.

Reed, W. M., J. M. Oughton, D. J. Ayersman, S. F. Giessler, and J. R. Ervin: 1995, 'Computer experience and learning style: Lynear versus nonlinear navigation'. *ED-MEDIA'95 - World conference on educational multimedia and hypermedia*, Charlottesville, pp. 802.

Rivlin, E., R. Botafogo, and B. Shneidermann: 1994, 'Navigating in hyperspace: designing a structure-based toolbox'. *Communications of the ACM* **37** (2), 87-96.

Schwarz, E., P. Brusilovsky, and G. Weber: 1996, 'World-wide intelligent textbooks'. *ED-MEDIA'96 - World conference on educational multimedia and hypermedia*, Boston, MA.

Shibata, Y. and M. Katsumoto: 1993, 'Dynamic Hypertext and knowledge agent systems for multimedia information'. *ACM Hypertext'93*, Seattle, WA, pp. 82-93.

Thomas, C. G.: 1995, 'Basar: A framework for integrating agents in the World Wide Web'. *IEEE Computer* **28** (5), 84-86.

Thomas, C. G. and G. Fischer: 1996, 'Using agents to improve the usability and usefulness of the World-Wide Web'. *Fifth International Conference on User Modeling, UM-96*, Kailua-Kona, Hawaii, pp. 5-12.

Tomek, I., H. Maurer, and M. Nassar: 1993, 'Optimal presentation of links in large hypermedia systems'. *ED-MEDIA'93, World conference on educational multimedia and hypermedia*, Orlando, FL, pp. 511-518.

Vassileva, J.: 1994, 'A practical architecture for user modeling in a hypermedia-based information system'. *4-th International Conference on User Modeling*, Hyannis, MA, pp. 115-120.

Vassileva, J.: 1996, 'A task-centered approach for user modeling in a hypermedia office documentation system'. *User Models and User Adapted Interaction* **6** (this issue).

Waterworth, J. A.: 1996, 'A pattern of islands: exploring public information space in a private vehicle'. In: P. Brusilovsky, P. Kommers and N. Streitz (eds.): *Multimedia, Hypermedia and*

Virtual Reality: Models, Systems, and Applications. Lecture Notes in Computer Science, Berlin: Springer-Verlag, pp. 266-279.

Yetim, F.: 1993, 'User-adapted hypertext explanation'. In: T. Grechenig and M. Tscheligi (eds.): *Human-Computer Interaction*. Berlin: Springer-Verlag, pp. 348-358.

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

Zhao, Z., T. O'Shea, and P. Fung: 1993, 'Visualization of semantic relations in hypertext systems'. *ED-MEDIA'93, World conference on educational multimedia and hypermedia*, Orlando, FL, pp. 556-564.

Zukerman, I. and R. McConachy: 1993, 'Consulting a user model to address a user's inferences during content planning'. *User Models and User Adapted Interaction* **3** (2), 155-185.

Zyryanov, M.: 1996, 'Adaptive local maps in the hypermedia components of intelligent learning environments'. In: P. Brusilovsky, P. Kommers and N. Streitz (eds.): *Multimedia, Hypermedia and Virtual Reality: Models, Systems, and Applications*. Lecture Notes in Computer Science, Berlin: Springer-Verlag, pp. 306-310.