8

Adaptive Navigation Support

Peter Brusilovsky

School of Information Sciences
University of Pittsburgh, Pittsburgh PA 15260
peterb@mail.sis.pitt.edu

Abstract. Adaptive navigation support is a specific group of technologies that support user navigation in hyperspace, by adapting to the goals, preferences and knowledge of the individual user. These technologies, originally developed in the field of adaptive hypermedia, are becoming increasingly important in several adaptive Web applications, ranging from Web-based adaptive hypermedia to adaptive virtual reality. This chapter provides a brief introduction to adaptive navigation support, reviews major adaptive navigation support technologies and mechanisms, and illustrates these with a range of examples.

8.1 Introduction

Adaptive hypermedia [9] is a research area at the crossroads of hypermedia and user modeling. Adaptive hypermedia systems (AHS) offer an alternative to the traditional "one-size-fits-all" hypermedia and Web systems by adapting to the goals, interests, and knowledge of individual users as they are represented in the individual *user models*. This chapter is focused on *adaptive navigation support* technologies originally developed in the field of adaptive hypermedia. By adaptively altering the appearance of links on every browsed page, using such methods as *direct guidance*, *adaptive ordering*, *link hiding and removal*, and *adaptive link annotation*, these technologies support personalized access to information. Over the last 10 years, adaptive navigation support technologies have been used in many adaptive Web systems in a range of application areas from e-learning to e-commerce. The evaluation of these technologies has demonstrated their ability to allow users to achieve their goals faster, reduce navigational overhead, and increase satisfaction [7; 18; 50; 52; 73].

After a brief introduction to the history of adaptive navigation support, this chapter offers a state-of-the-art overview of adaptive navigation support. The overview is divided into two parts. The first part focuses on adaptation technologies and attempts to answer the question: *What kind of adaptation effects may be useful to provide guidance to the users of Web hypermedia systems?* The second part focuses on adaptation mechanisms and attempts to answer the question: *How can these adaptation effects be produced?* Both parts are illustrated with a range of examples. The last section discusses the prospects of extending adaptive navigation support beyond Web hypermedia.

8.2 Adaptive Navigation Support: From Adaptive Hypermedia to the Adaptive Web

Research on adaptive navigation support in hypermedia can be traced back to the early 1990's. By that time, several research teams had recognized standard problems found in static hypertext within different application areas, and had begun to explore various ways to adapt the behavior of hypertext and hypermedia systems to individual users. A number of teams addressed problems related to navigation in hypermedia—such as the problem of inefficient navigation or the problem of being lost—which had been discovered when the field of hypertext reached relative maturity at the end of the 1980's [46]. Within a few years, a number of navigation support technologies were proposed [4; 19; 33; 52]. While the proposed technologies were relatively different, they shared the same core idea: within a hypertext page (node), adapt the presentation of links to the goals, knowledge, and preferences of the individual user. The adaptive navigation support technologies introduced by early adaptive hypermedia systems were later classified as *direct guidance*, *sorting*, *hiding*, *annotation,* and *map adaptation* [8]. Most of these systems used adaptation mechanisms based on manual page indexing and provided navigation support within a closed corpus of documents.

The Web as "hypermedia for everyone" immediately provided an attractive platform for adaptive hypermedia applications. The problem of navigation support in Web hypermedia attracted many new researchers to the field. A good number of these researchers were motivated by pre-Web adaptive hypermedia and focused on exploring a set of known adaptive hypermedia technologies in the new Web context. Other researchers suggested new techniques such as link *disabling* and *generation* [9]. Several new adaptation mechanisms were explored including *content-based* and *social* mechanisms that allowed navigation support in an open corpus. As the Web has developed, the focus of work has also moved from exploring isolated techniques using "lab-level" systems to developing and exploring "real world" systems for different application areas such as e-learning, e-commerce, and virtual museums.

Altogether, pre-Web and Web-based AHS with adaptive navigation support explored a broad range of adaptation technologies and mechanisms in many application areas. The knowledge of these technologies and mechanisms and their effectiveness is important for the developers of future adaptive Web systems. The next two sections attempt to summarize this knowledge, presenting the most popular adaptation technologies and mechanisms, and pointing out relevant empirical studies.

8.3 Adaptive Navigation Support: Adaptation Technologies

8.3.1 Direct guidance

Direct guidance is the simplest technology for adaptive navigation support. Direct guidance suggests the "next best" node (or sometimes, several alternative nodes) for the user to visit according to the user's goals, knowledge, or/and other parameters that have been represented in the user model. On the interface level, direct guidance can be presented to the user in two main forms. If the link to the suggested node is already present on the page, it can be outlined or emphasized in some other way. For example, WebWatcher [1] and Personal WebWatcher [63] indicated the recommended link(s) by a pair of icons showing curious eyes (Fig. 8.1). Alternatively, the system can generate a dynamic "next" link which is connected to the "next best" node.

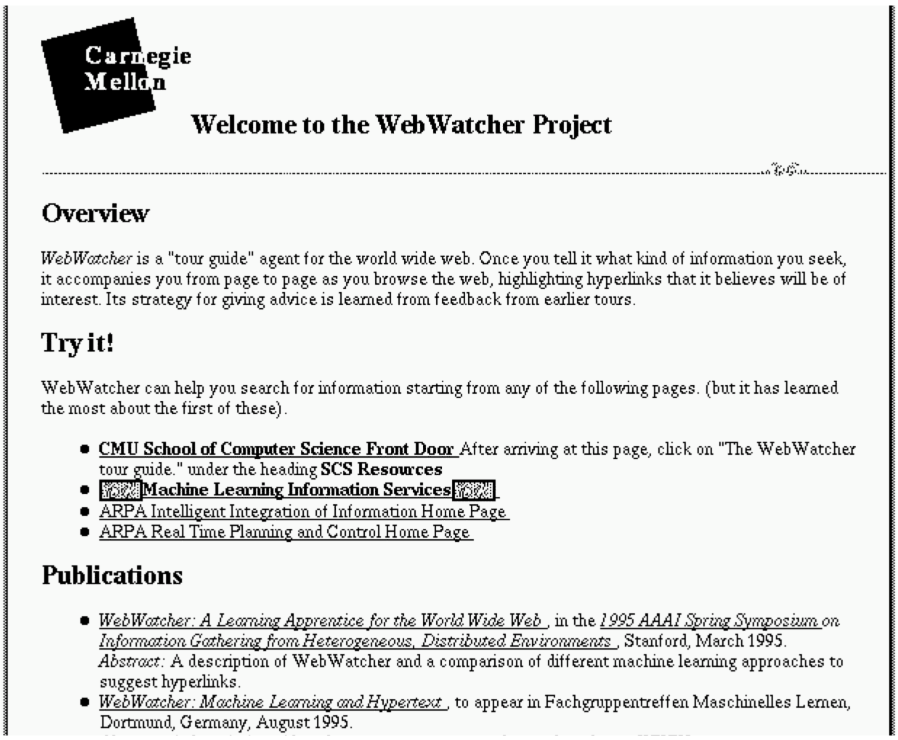


Fig. 8.1. Direct guidance in Personal WebWatcher. The recommended link (second from the top) is outlined by a pair of “curious eyes” icons. Used with permission from the author [63].

A known problem with direct guidance is that it provides no support for users who don’t wish to follow the system's suggestions. Due to this problem, although direct guidance was popular in the early days of adaptive hypermedia, it is now mostly replaced by other navigation support technologies, which will be introduced below. The only group of systems where this approach remains popular are adaptive educational hypermedia systems, especially those that have roots in Intelligent Tutoring Systems such as HyperTutor [69], ELM-ART [78], or InterBook [14]. In this group, direct guidance became the hypermedia form of the traditional *curriculum sequencing* mechanisms. Several studies reviewed in [10] demonstrated that novice users with poor domain knowledge have problems in dealing with alternative navigation choices and can be best supported by direct guidance technology.

8.3.2 Link Ordering

The idea of an *adaptive sorting* or *ordering* technology is to prioritize all the links of a particular page according to the user model and some user-valuable criteria: the closer to the top, the more relevant the link is. While adaptive sorting was first introduced in 1990 in the Hypadapter system [49], the most frequently referred example of this technology is HYPERFLEX [52]. HYPERFLEX attempts to order links from the current page to related pages according to the user-perceived relevance of these pages to the current one. If the user thinks that the presented order is incorrect, the links can be manually reordered by dragging. Manual link reordering is considered by the system as a means of relevance feedback and is used to update the user model. If the user selects the current search goal from the list of existing goals (new goals can also be introduced), link ordering on every page also takes into account link relevance to the selected goal. Most important to the HYPERLEX work was not the specific adaptation technology, but rather the study of the user's link ordering, which was reported in the same paper [52]. The study demonstrated that adaptive link ordering significantly reduces navigation time and the number of steps that are required to locate the information that the user is looking for. These results helped to attract attention to link ordering and adaptive navigation support in general. It should be noted, though, that time reduction is not exclusively limited to sorting technologies. Similar time/steps reduction was later observed for other navigation support technologies, such as link hiding and annotation [18; 65] and is currently considered to be one of the most important values of adaptive navigation support in general.

Despite its demonstrated effectiveness, link sorting has not become very popular, due to its limited applicability. As shown in Table 8.1, it can be used for non-contextual links, but is difficult to use for an index page or a table of contents (which usually have a predefined order of links), and can never be used with contextual links or maps. Another problem with adaptive ordering is that this technology makes the order of links unstable: it may change each time the user enters the page. Since the first introduction of link sorting, several user studies have demonstrated that unstable order of options in menus and toolbars creates problems for at least some categories of users [34; 53]. As a result, this technology is presently used in only a few contexts where the unstable order of links creates no problem.

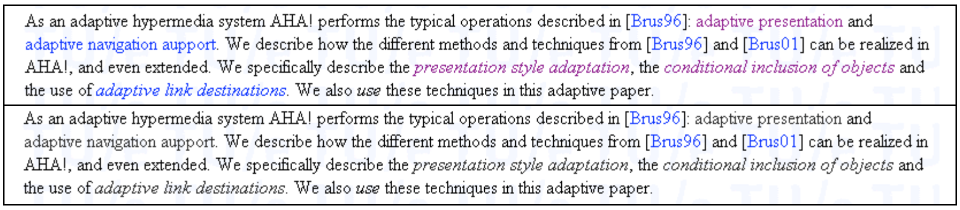
One such beneficial context is adaptation of link order to long-term user characteristics. In this context, different users may see a different order for links, but it is stable for each user for the whole time they are working with the system. For example, several adaptive e-learning systems order links to the different educational resources available for a topic according to the relevance of these resources to the user's learning style [55].

Another appropriate context includes several kinds of system where all or some pages have an unstable set of links. Since the set of links on a page is not fixed, a stable order does not exist anyway. In this situation the "conceptually stable" ordering offered by link sorting can become an attractive solution. Good examples of this may be found among adaptive news systems reviewed in Chapter 18 of this book [3] and collaborative resource gathering systems such as CoFIND [39] or COMTELLA [26]. Adaptive news systems typically present links to recommended news articles in a single list or on several pages by category. This list is unstable because new articles are constantly added and old articles removed. In this context, it is very natural to sort the links according to the modeled interests of the user. This ordering is typically performed by content-based mechanisms.

In collaborative resource gathering systems, users collect useful Web resources by adding interesting links to topics. Each topic may have a short introduction and a collection of links that is unstable by its nature (since resources are constantly added and even sometimes removed). To present these links, the cited systems use social mechanisms to sort topic links according to the perceived community interests. For similar reasons, link sorting is frequently used in combination with link generation (see section 8.3.6 and Fig. 8.10 for examples of this combination).

8.3.3 Link hiding

The purpose of navigation support by *hiding* is to restrict the navigation space by hiding, removing, or disabling links to irrelevant pages. A page can be considered irrelevant for several reasons: for example, if it is not related to the user's current learning goal or if it presents materials which the user is not yet prepared to understand. Hiding protects users from the complexity of the whole hyperspace and reduces their cognitive overload. Educational hypermedia systems have been the main application area where adaptive hiding techniques have been suggested and explored. Indeed, beginning with just a part of the whole picture then introducing other components step by step as the student progresses through the course is a popular educational approach and adaptive hiding offers a simple way to implement this. Early adaptive hypermedia systems used a very simple method of hiding links—essentially removing the link as well as the anchor from a page. A good example is the ISIS-Tutor educational hypermedia system [18], which shows very few links when the student begins to work with the system but gradually makes more and more links visible, reacting to the growth of the student's knowledge of the subject. De Bra and Calvi [29] later called the ISIS-Tutor approach *link removal* and have suggested several other variants for link hiding based on the separation of three features of a link: the anchor, the visible indication, and the functionality. For example, link *hiding* preserves the link anchor (hot word), but removes all visual indications that it is a link (i.e., blue color and underline). Link *disabling* removes the functionality, i.e., the ability of the link to take the user to the related page. Both technologies (as well as their combination) extend the applicability of link hiding to contextual links where the anchor simply can't be removed. An example of link hiding in De Bra's AHA! framework is shown on Fig. 8.2. This example is taken from their adaptive paper, which presented their framework [31]. A number of studies of link hiding revealed that it is best used as a "unidirectional" technology. While gradual link enabling as used in ISIS-Tutor has been acceptable and effective, the reverse approach has been found questionable: users become very unhappy when previously available links become invisible or disabled.



As an adaptive hypermedia system AHA! performs the typical operations described in [Brus96]: adaptive presentation and adaptive navigation support. We describe how the different methods and techniques from [Brus96] and [Brus01] can be realized in AHA!, and even extended. We specifically describe the *presentation style adaptation*, the *conditional inclusion of objects* and the use of *adaptive link destinations*. We also *use* these techniques in this adaptive paper.

As an adaptive hypermedia system AHA! performs the typical operations described in [Brus96]: adaptive presentation and adaptive navigation support. We describe how the different methods and techniques from [Brus96] and [Brus01] can be realized in AHA!, and even extended. We specifically describe the *presentation style adaptation*, the *conditional inclusion of objects* and the use of *adaptive link destinations*. We also *use* these techniques in this adaptive paper.

Fig. 8.2. Link hiding in AHA! framework taken from the adaptive paper [31]. The upper fragment shows several links leading to other sections of the paper. On the lower fragment these links are hidden—the purple color indicating the presence of a link is replaced by the black color of the surrounding text.

8.3.4 Link Annotation

The idea of *adaptive annotation* technology is to augment the links with some form of annotation, which lets the user know more about the current state of the nodes behind the annotated links. These annotations are most often provided in the form of visual cues. Manuel Excel [33] introduced link annotation with different icons, ISIS-Tutor [17] changed the color and intensity of the anchors, and Hypadapter [49] explored altering anchor font sizes. The Web generation of adaptive hypermedia systems introduced several kinds of verbal annotations that could be shown next to the anchor [45], on the browser's status bar [14], or as a *gloss* that popped up when the user moused over a link [82]. All of these approaches to link annotation are now in use, but the most popular are probably icon-based annotation and mouseovers. Naturally, annotation can be used with all possible forms of links. This technology preserves a stable order to the links, thus avoiding problems with incorrect mental maps. Annotation is generally a more powerful technology than hiding: hiding can distinguish only two states for related nodes—relevant and non-relevant—while the currently existing annotation applications can distinguish up to six states. For all the above reasons, adaptive annotation has grown into the most frequently used adaptive navigation support technology.

Some of the benefits of adaptive link annotation have been explored in several studies. For example, an early study of the ISIS-Tutor system [18] compared three versions of the ISIS-Tutor: non-adaptive, adaptive annotation, and a combination of both adaptive hiding and annotation. The results of the study demonstrated that the same educational goal is achieved with either of the adaptive versions with much less navigational overhead than with the non-adaptive version. The overall number of navigation steps, the number of unforced repetitions of previously studied concepts, and the number of task repetitions (i.e., trials to solve a previously visited task) were significantly smaller for both adaptive versions.

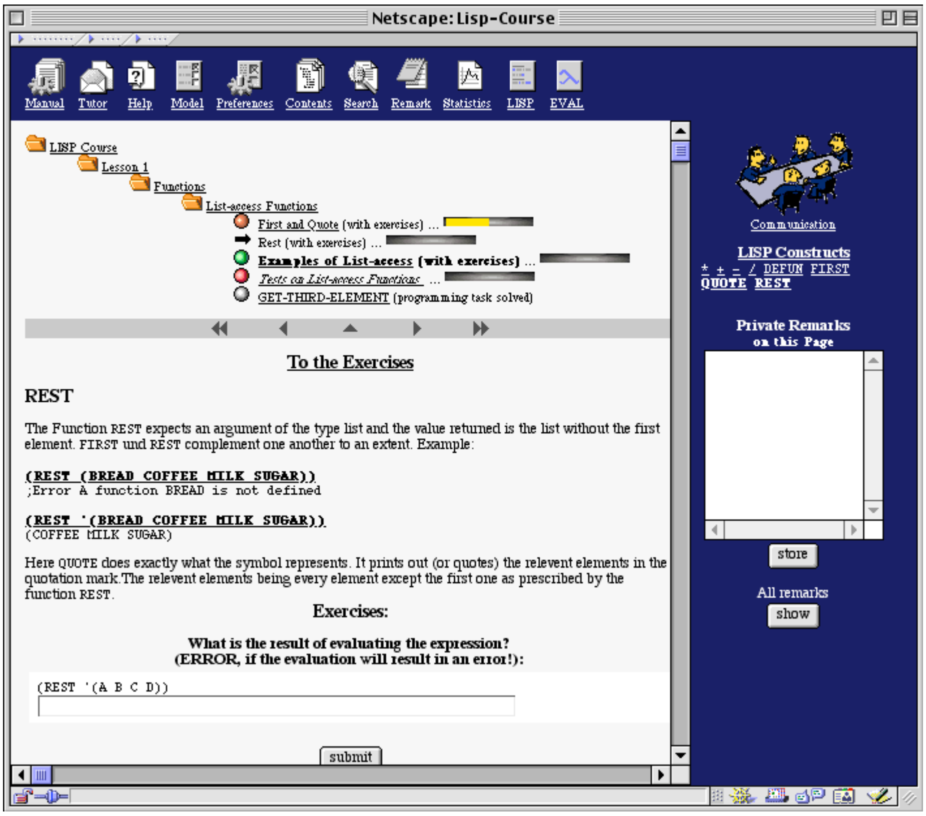


Fig. 8.3. Adaptive navigation support in ELM-ART, an electronic textbook for learning LISP. Adaptive annotation in the form of colored bullets (a traffic light metaphor) shows the educational state of pages behind the links. Adaptive annotation in the form of progress bars visualizes the student's demonstrated level of knowledge of related concepts.

A popular example of adaptive annotation in Web hypermedia is ELM-ART [20], which was one of the first Web-based systems with adaptive navigation support. ELM-ART introduced the traffic light metaphor for adaptive navigation support in educational hypermedia. In this metaphor, a green bullet in front of a link indicates recommended readings, while a red bullet indicates that the student may not be able to understand the information behind the link yet. Other colors, like yellow or white, indicate more educational states such as the lack of new knowledge behind the link. This kind of annotation is produced by an indexing-based mechanism and will be explained in more details in section 8.4.4. In addition to link annotation, ELM-ART also supports direct guidance. Fig. 8.3 shows adaptive annotation in the most recent version of ELM-ART [78]. This version augments the traffic-light annotation (which indicates the educational status of a page) with progress-based annotation (which indicates the level of user knowledge for a LISP concept associated with this page). A combination of these two kinds of annotations is currently very popular in adaptive educational hypermedia. A study of ELM-ART [78] demonstrated that casual users stay longer within a system when adaptive navigation support is provided. The study also provided evidence that direct guidance works best for users with little previous knowledge while adaptive annotation is most helpful for users with a reasonable amount of subject knowledge.

8.3.5 Link Generation

Link generation is the "newest" adaptive navigation support technology. There has been little need to introduce link generation in the context of pre-Web adaptive hypermedia with its small, well-linked, closed corpus document collections. This technology was introduced in several early adaptive Web systems in 1996 [14; 78; 81] and became very popular in Web hypermedia with its abundance of resources. Unlike classic annotation, sorting or hiding technologies that adapt the presentation of *pre-authored* links, link generation actually creates new, non-authored links on a page. There are three known kinds of link generation: (1) discovering new, useful links between the documents and adding them permanently to the set of existing links; (2) generating links for similarity-based navigation between items; and (3) the dynamic recommendation of links that are useful within the current context to the current user (i.e., the current goal, knowledge, or interests, as reflected in the user model). The first two kinds of link generation are typically non-adaptive. We should mention, however, several known projects that explored creating new links for a group of users as a result of an analysis of group navigation patterns [5; 59; 81] and a few attempts to develop adaptive similarity-based navigation [23]. The third technology is naturally adaptive, since link generation is driven by the user's profile and context.

Since link generation is now very popular in several kinds of adaptive Web-based systems, this section is a good place to comment on the similarities and differences between using this technology in adaptive navigation support systems and the various Web recommender systems that are presented in chapters 9, 10, 11, and 12 of this book [24; 68; 71; 72]. Recommender systems attempt to suggest a list of items that are relevant to the user's short- or long-term interests. These items may or may not be part of a hyperspace. If they are, the recommendation can be presented as a set of generated links. Even those systems that attempt to recommend items in hyperspace typically do not take the current user location in hyperspace (context) into account, and instead offer links that should be of interest in general. On the other hand, navigation support systems focus on helping users to find their way through hyperspace by adapting links on a page. Link adaptation can take into account various features of the user and may take many forms as well, including, as a specific case, link generation adapted to the user's interests. In all cases, navigation support techniques provide guidance that takes into account the user's current location in hyperspace. So, when guidance is provided by link generation, a navigation support system attempts to introduce additional links that may be useful in the current context. Since navigation support systems focus on the interface and recommender systems focus on the underlying technology, the difference between these two groups is not clear-cut. Evidently, a small class of systems that generates links according to the user's interests and takes into account the user's current location can be classified as both a Web recommender system and an adaptive navigation support system. A well-known example of this is Amazon.com (http://amazon.com). This system recommends links to products that were considered or purchased by other users who viewed the current product.

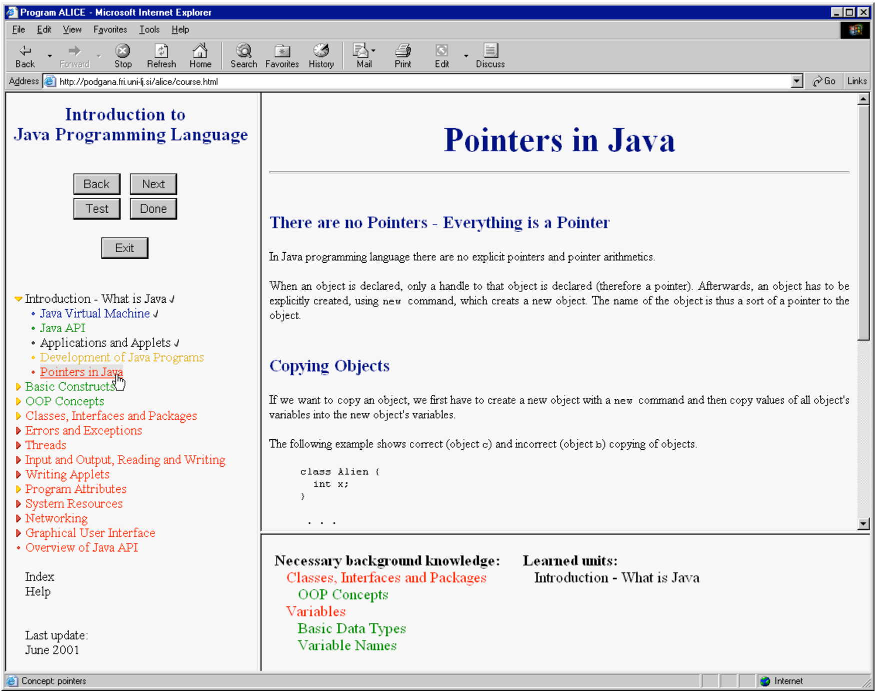


Fig. 8.4. Link generation and link annotation in ALICE. Follow-up links are generated in the bottom right frame in three groups - next possible units, necessary background units, and all learned units. The example on the figure doesn't suggest next possible units since the current unit "Pointers in Java" is not yet ready to be learned (note that it is annotated with red color on the table of contents in the left frame).

A good example of link generation adapted to user knowledge is ALICE [54], an electronic textbook about the Java programming language. ALICE includes 13 chapters and 97 sections devoted to different Java concepts and uses link generation as the main navigation support approach. There are no stable links between sections; instead, the links are generated dynamically according to the current user level of knowledge. These dynamically-generated links are added to the end of the viewed section in three groups—next possible units, necessary background units, and all learned units (Fig 8.4). The system uses a sophisticated approach to model the user's knowledge of Java, which is reviewed in more detail in Chapter 1 of this book [16]. The evaluation of navigation support in ALICE revealed that students who follow the generated navigation suggestions score better on tests.

8.3.6 Comparing and Combining the Technologies

The link adaptation technologies reviewed above have a lot in common, since they are motivated by the same need, guiding the user in hyperspace. At the same time, these technologies are quite different in their applicability. Part of this is due to the technical applicability of each specific technology for adapting different kinds of links. Hypertext links (i.e., visible and "clickable" representations of the related pages to which the user can navigate) can be classified in several groups (Table 8.1):

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 cannot be removed from it. These links and the corresponding anchors, can be annotated or disabled, but cannot be sorted or completely hidden.

Local non-contextual links. This type includes all kinds of links on regular hypermedia pages, which are not embedded in the context 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, removed, generated, or annotated, although disabling or hiding this kind of links (with the anchor preserved) makes little sense.

Links from index and table of contents. An index or a table of contents page can be considered to be a special kind of page, which contains only links that are organized in a specific order (content order for content pages and alphabetic order for index pages). As a rule, links from index and content pages are non-contextual, yet these links can't be sorted and application of all hiding technologies in this context has questionable usability.

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 by merely clicking on a representation of the desired node. From a navigation point of view, these clickable representations of nodes are navigational links, while paradoxically, the arrows serving as a representation of links are not used for direct navigation.

In brief, the analysis of technical applicability demonstrates that some technologies have much wider applicability than others. It is not surprising that the most universal technologies—annotation and generation—are also currently the most popular. However, there is also another aspect to the applicability: A range of studies of adaptive navigation support systems indicates that the effect of a specific technology may be different for different classes of users. For example, a number of studies provide evidence that direct guidance is beneficial to users with a low level of domain knowledge, while link annotation works best with users who are already above the starting level of knowledge [10]. The applicability of different technologies is important to consider when developing adaptive navigation support systems.

In addition to the applicability limits, different technologies may be best suited for the different needs of an adaptive system. As a result, we see fewer and fewer “purist” systems that use exactly one of the technologies. The majority of practical systems use different technologies in parallel or in different parts of the system. For example, among the systems already mentioned above, ISIS-Tutor uses direct guidance, hiding, and annotation; Hypadapter uses sorting, hiding, and annotation; AHA! uses hiding and annotation; ALICE uses generation and annotation, and both InterBook and

ELM-ART use direct guidance, annotation, and generation. Sometimes different technologies used in the same system are based on different mechanisms, but more frequently the same mechanism powers all adaptation technologies in a system. An example of using an index–based mechanism to produce direct guidance, annotation, and generation in InterBook is reviewed in section 8.4.4.

Table 8.1. Adaptive navigatßion support technologies and their applicability.

	Direct guidance	Sorting	Hiding	Annotation	Generation
Contextual links	OK		Disabling	OK	
Non-contextual links	OK	OK	OK	OK	OK
Table of contents	OK			OK	
Index	OK			OK	
Hyperspace maps	OK		OK	OK	

8.4 Adaptation Mechanisms for Adaptive Navigation Support

8.4.1 Simple Adaptation Mechanisms

To make the presentation complete, we must start with simple adaptation mechanisms that do not require advanced adaptation algorithms and yet can be of real use in a range of contexts. The most popular examples are history-based and trigger-based mechanisms

History-based mechanisms. History-based mechanisms simply count how many times each node in the hyperspace is accessed and attempt to represent this information visually. The oldest example is the rendering of visited links in an alternative color—a feature of every Web browser since Mosaic times (and actually inherited from hypertext research). Early research on adaptive navigation support attempted to extract more value from the stored history. For example, the MANUEL EXCEL system [33] dynamically annotated hypertext links with three different icons (a clear, gray, or black magnifying lens) to express the extent to which the area of hyperspace behind each link had previously been visited by the user (Fig. 8.5). Experiments with the system provided early evidence in favor of adaptive link annotation.

Trigger-Based Mechanisms. A trigger-based mechanism can be considered as an extension of a simple history-based adaptation. The idea of trigger-based adaptation is to connect a link with some simple event. Once this event has happened, the state of a binary trigger associated with a link is changed, resulting in a changed link appearance. A number of Learning Management Systems such as TopClass [77] use the simple trigger-based mechanism to control student access to learning content. A link to a section with learning content can be disabled or enabled at a specified time or after a specific quiz is completed by the user with a score under or above a threshold. A combination of these triggers allows teachers to provide some amount of class-level and individual personalization.

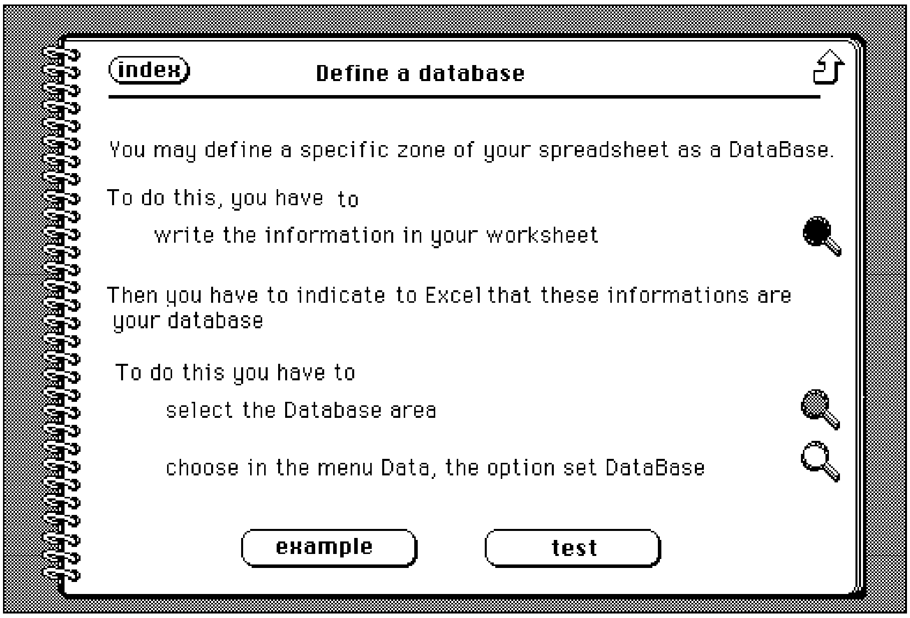


Fig. 8.5. Annotations for topic states in MANUEL EXCEL: not seen (clear magnifying lens), partially seen (grey lens), and completed (black lens).

Progress-Based Mechanisms. The power of simple history-based mechanisms can be expanded if the adaptive system is able to track user visit to a page on a deeper level. For example, an information system may track time spent reading a page [64] or amount of page exploration (using eye-tracking or mouse tracking). Educational systems can measure the success of user work, e.g., a quiz that a link leads to can be solved partially, completely, or not yet attempted. The progress can be shown graphically next to each link to pages with educational activities helping the user to decide whether to visit these pages or not. The use of information about the hypertext structure can further expand the power of progress-based adaptation. For example, in a hierarchically organized hyperspace, progress can be propagated up the hierarchy. Visual presentation of user progress for the top-level hyperspace topics provides an easy-to-grasp overview of the current state of work.

An example of using a progress-based mechanism with propagation in an educational context is provided by QuizGuide [21]. This system attempts to guide students to the most relevant self-assessment quizzes. Quizzes are grouped into topics. Once a topic link is "expanded," the links to all topic quizzes become available. Adaptive navigation support is provided on the topic level. The system traces correct and incorrect answers for all questions, calculating mastery levels for each quiz. These levels are propagated to the topic level, forming the mastery view of the whole topic. The icon annotating the link to the topic expresses this mastery in a target-arrow metaphor: the more arrows, the higher the level of mastery achieved for the topic (Fig. 8.6). These annotations allow students to see which topics are sufficiently mastered and which require additional work. The color of the target in QuizGuide attempts to express how important it is to attend to the topic, from the perspective of the class schedule. Current topics are marked by bright blue targets, their prerequisites by light blue targets, and other past topics by gray targets. Topics that are not yet introduced in class are crossed out, suggesting that the student is not ready to attempt them. This kind of annotation is supported by a trigger-based mechanism controlled by the teacher through the class schedule. The evaluation of progress-based navigation support in QuizGuide demonstrated that this technology has succeeded in guiding the user to the most appropriate quizzes (as demonstrated by an increased rate of correct answers). In addition, the provision of adaptive visual cues significantly increased user motivation to work with the system, more than doubling the amount of non-mandatory work with the self-assessment quizzes that the students were willing to do [22].

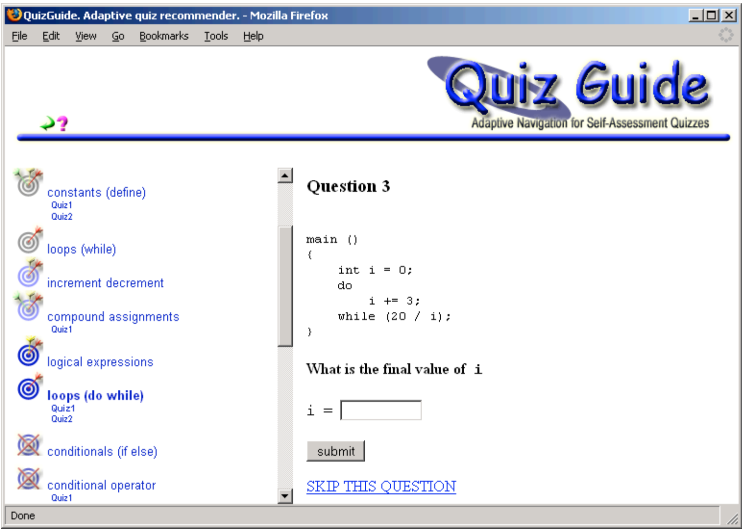


Fig. 8.6. Progress-based adaptive navigation support in QuizGuide. Depending on the percentage of correct answers to questions belonging to a topic, the icon annotating the link to the topic shows from zero to three arrows.

8.4.2 Content-Based Mechanisms

Content-based adaptive navigation support mechanisms make a decision whether to suggest the user a path to a specific page by analyzing page content. Most of these mechanisms process pages to obtain keyword vectors and compare them with the profile of user interests. Link following is treated as an expression of user interests and is used for updating the user profile. More information on user profiles and document modeling can be found in Chapters 2 and 5 of this book [44; 60].

Content-based approaches were rarely used in pre-Web hypermedia. Interest in this area was attracted by the development of several pioneer systems in 1995-1996, such as WebWatcher [1], Letizia [58], Syskill & Webert [67], and Personal WebWatcher [63]. These systems influenced a number of more recent projects on both Web recommenders and content-based navigation support. While some of the pioneer systems with content-based navigation support were applied in the closed corpus context (i.e., a single Web site), others clearly demonstrated the most important innovation of content-based approaches: the ability to work with the open corpus Web. This idea was most clearly spelled out in the Letizia system, which was designed as an agent assisting user browsing by “running ahead” of the user, checking the content of pages behind the links, and suggesting the most relevant links to follow.

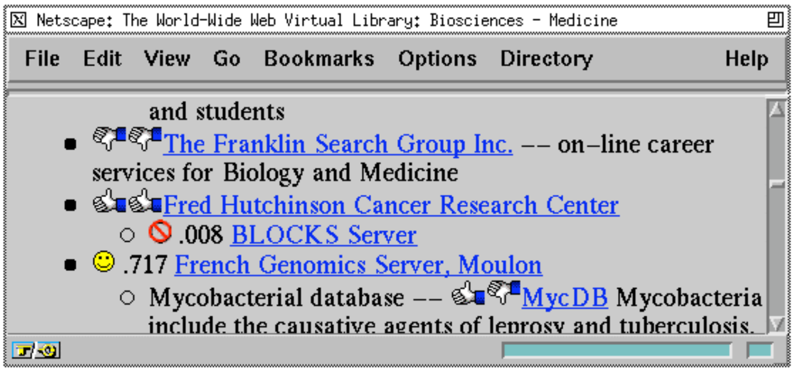


Fig. 8.7. Content-based navigation support in Syskill & Webert. Thumb icons identify pages that were previously rated, smiley icons point to a potentially interesting but not yet visited page. Used from [67] with author’s permission.

It is probably due to this common root that a number of systems with content-based navigation use essentially the same decision-making mechanisms as content-based recommender systems. A review of these mechanisms is provided in Chapter 10 of this book [68]. A good example of the application of content-based recommender approaches in the context of navigation support is provided by the Syskill & Webert system [67]. Syskill & Webert attempts to learn user interests related to several topics while assisting user browsing. To provide relevance feedback to the system, the user explicitly rates encountered pages as hot, cold, or lukewarm. User ratings along with page representations as a bag-of-words are used to build a profile of user interests on different topics. As soon as the topic profile is discovered, the system starts suggesting interesting links on the current page by pre-fetching pages behind the links and classifying them according to the profile. Navigation support is provided by link annotation, i.e., links annotated with icons. Several different icons allow the user to differentiate previously rated pages and new, potentially interesting pages (Fig. 8.7). The prefetching-classification-annotation approach suggested in Syskill & Webert is straightforward, powerful, and universal. It could be used to recommend links on any page, whether a regular hypertext page with embedded links, a generated page with recommended links, or a page with links returned by a search engine in response to a query. Syskill & Webert demonstrated this flexibility by providing link annotation on a page generated by the Lycos search engine.

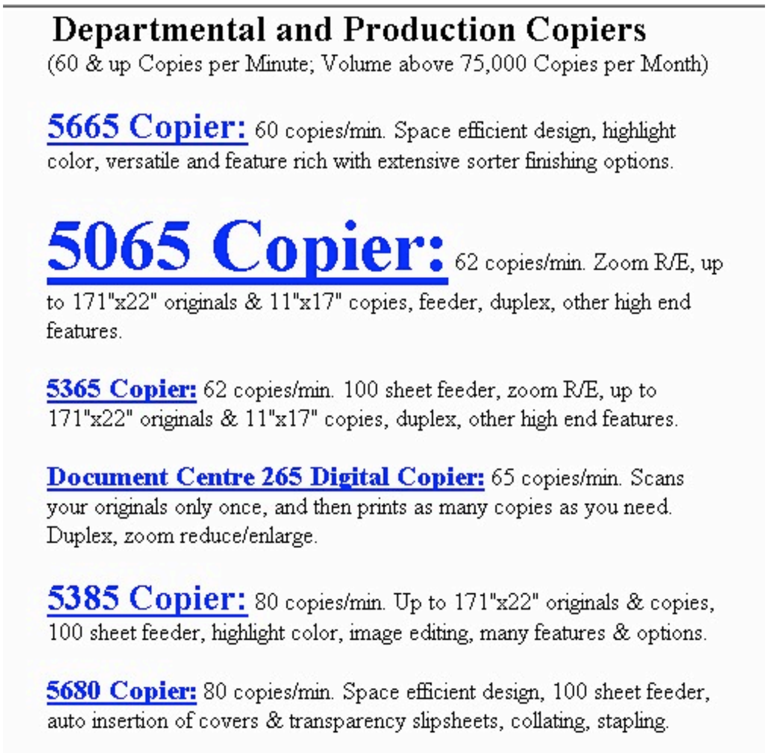


Fig. 8.8. Content-based navigation support in ScentTrails. The size of the link font indicates how relevant the region of hyperspace behind this link is to the user's search goal. Used from [65] with the author's permission.

An example of content-based navigation support that differs from Syskill & Webert in several aspects is ScentTrails [65]. To start with, this system adapts to the user's search goal (formulated as a query), not to user interests. While this system also applies link annotations, it uses font size, not icons, in order to express more levels, when judging the relevancy of the path started by this link to the user goal (Fig. 8.8). However, most importantly, ScentTrails demonstrates the ability to look more than one step ahead when guiding users in hyperspace. The size of the link font shows the cumulative relevance of a whole region of hyperspace behind this link, i.e., a larger link font may indicate that the relevant page is not directly behind the link, but several steps ahead, on the path started by this link. The system is able to generate this advanced level of guidance by taking into account not only page content but also links between pages. The mechanism used in ScentTrails is based on the idea of an *information scent*. The simple scent of a page is its relevance to the user goal (a query) that is calculated using traditional information retrieval techniques. The full scent of pages in a connected set is calculated by propagating simple scents along the links. The assumption is that scent emanates equally from a page along each of its links, but decreases on each iteration. Potentially, this approach can work in the open corpus context by calculating information scent on the fly, but it is very time consuming due to the large number of pages that have to be processed. To keep response time small, the authors suggested a scent-calculating approach that is based on a relevance matrix recursively computed in advance. This effectively restricted the scope of this approach to a closed corpus context, such as a Web site. The evaluation of ScentTrails demonstrated that a full-scent version of the system allowed the user to achieve their goal significantly faster and with a much higher rate of success.

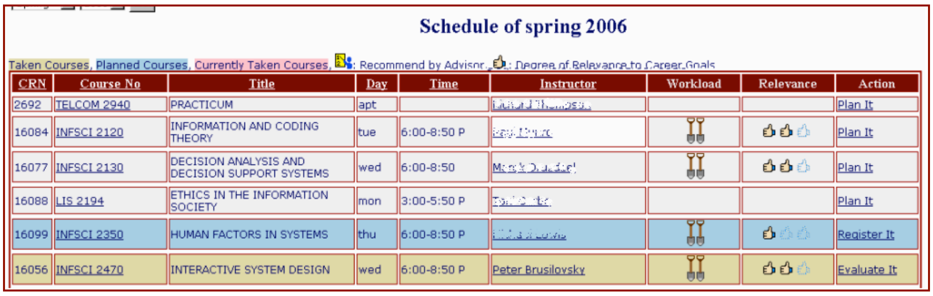
8.4.3 Social Mechanisms

Social mechanisms are based on the idea of *social navigation,* which capitalizes on the natural tendency of people to follow the direct and indirect cues of the activities of others, e.g., going to a restaurant that seems to draw many customers, or asking others what movies to watch. Social navigation in information space was originally introduced by Dourish and Chalmers as "moving towards clusters of people" or "selecting subjects because others have examined them" [38]. Social navigation support can be offered in a *direct* or *indirect* form. Direct social navigation means the direct interaction of users with each other in an information space. Indirect social navigation traces the activities of the community of users in the information space to guide new users in the system.

A typical approach to implement direct social navigation in hyperspace is to annotate links to pages that are currently being visited by other users with special icons. Several projects suggested technical solutions on how to augment links with this information [2]. Once visiting the same page, the users can typically communicate with each other. An elaborate implementation of this approach, using link annotation on a document map, was implemented in the EDUCO system [56].

Systems with indirect social navigation are typically classified into two groups: *history-enriched environments* and *collaborative filtering systems* [36]. History-enriched environments provide support for navigating through an information space by making the aggregated or individual action of others visible. This form is predominantly used by social navigation support mechanisms. The term history-rich information space was introduced by Wexelblat and Maes who implement this concept in their Footprints system [80], which visualizes usage paths throughout a web site. With the Footprints system, new users can see the popularity of each link on the current page and make navigation decisions. This approach is based on counting user passage through a link or user visits to a page and is known as a *footprint-based approach*. It was later implemented in several other systems such as CoWeb [37] and the first version of Knowledge Sea II [11]. A more recent version of the Knowledge Sea II system [40; 41] extended the footprint-based approach and explored annotation-based social navigation support. The extended version of the footprint-based approach takes into account time spent reading each page in order to scale footprints left by incomplete and accidental page visits and to obtain more reliable evidence of this page's relevance to the community of users [40]. Annotation-based social navigation support creates a history-rich environment by visualizing page annotations made by a community of users. This system is presented in more detail in Chapter 22 of this book [15].

Collaborative filtering is a technique for providing recommendation based on earlier expressed preferences or the interests of similar users. Collaborative filtering mechanisms are frequently powered by explicit user ratings, although recent systems have explored the use of implicit interest indicators [28]. While collaborative filtering mechanisms are mostly used in collaborative Web recommender systems reviewed in Chapter 9 of this book [71], a few of systems used it for providing social navigation support. A straightforward example of navigation support based on community ratings is provided by collaborative resource gathering systems such as CoFIND [39] or COMTELLA [26], which were reviewed in section 8.3.2 above. A more elaborate example is shown by the CourseAgent system [42].


Schedule of spring 2006

Taken Courses, Planned Courses, Currently Taken Courses, : Recommend by Advisor, : Degree of Relevance to Career Goals

CRN	Course No	Title	Day	Time	Instructor	Workload	Relevance	Action
2692	TELCOM 2940	PRACTICUM	apt		[illegible]			Plan It
16084	INFSCI 2120	INFORMATION AND CODING THEORY	tue	6:00-8:50 P	[illegible]			Plan It
16077	INFSCI 2130	DECISION ANALYSIS AND DECISION SUPPORT SYSTEMS	wed	6:00-8:50	[illegible]			Plan It
16088	LIS 2194	ETHICS IN THE INFORMATION SOCIETY	mon	3:00-5:50 P	[illegible]			Plan It
16099	INFSCI 2350	HUMAN FACTORS IN SYSTEMS	thu	6:00-8:50 P	[illegible]			Register It
16056	INFSCI 2470	INTERACTIVE SYSTEM DESIGN	wed	6:00-8:50 P	Peter Brusilovsky			Evaluate It

Fig. 8.9. Social navigation support in the schedule view of the CourseAgent system. Thumbs-up icons express the predicted usefulness of the course for the student. Darker background colors (blue and gold) indicate previously taken or planned courses.

CourseAgent attempts to recommend relevant courses to graduate students in Information Science taking into account the ratings of users who already took these courses. To make recommendations more reliable, the system uses a taxonomy of career goals. Every user is expected to select several career goals. Every course is rated independently in regard to each career goal of the rater. To predict the usefulness of a course for a student with a specific set of career goals, the system integrates existing ratings of this course in regard to these career goals. Course ratings are presented to students through link annotation. Wherever a link to a useful course is shown in the system (i.e., in a course schedule for the current semester or in a course catalog), it is augmented with thumb-up icons. The number of icons (one to three) expresses the predicted usefulness of the course (Fig. 8.9). The system also applies simple history-based navigation support, using special background colors to mark previously taken (gold) or planned (blue) courses.

8.4.4 Indexing-Based Mechanisms

Indexing-based mechanisms are the most popular and powerful mechanisms for providing adaptive navigation support in adaptive hypermedia. The idea of the indexing-based approach is similar to that of the content-based approach: represent some information about each page that can be matched to the user model and used to make a decision about whether and how to provide guidance. The difference between these two appoaches come from the representation. Content-based mechanisms use automatically-produced word-level document representations (presented in Chapter 5 of this book [60]) and similar user profiles (presented in Chapter 2 of this book [44]). Indexing-based mechanisms use manually-produced concept-level document representation and concept-level overlay models (presented in Chapter 1 of this book [16]). Concept-level representation is more powerful and precise, but due to involved manual processing it is rather expensive, which limits the application of indexing-based mechanisms to the closed corpus context.

The concept-level page representation is produced by expressing the content of each page in terms of external concept-level models. It means that each page is connected (associated) to one or more concepts that describe some aspect of this page. This process is known as indexing, because specifying a set of underlying concepts for every page is similar to indexing a page with a set of keywords. To provide a match between page indexing and user models, the same external model must be used for both building an overlay user model and page indexing. In the majority of adaptive hypermedia systems, the external model used for indexing is simply a concept-level *domain model* introduced in Chapter 1 of this book [16]. However, a number of systems use different kinds of models for indexing, such as a hierarchy of tasks, a taxonomy of learning styles, etc. These models are reviewed as *generalized models* in Chapter 1 of this book [16]. Since the aspects of page representation by indexing are not covered anywhere else in this book, the following subsections provide a brief review of these indexing approaches. For simplicity, this section refers to the elements of the external models as *concepts* regardless of their nature. Following this review, we present an example of using the indexing-based approach in the InterBook system.

Classification of Indexing Approaches. There are three attributes that are important to distinguish different indexing approaches, from the adaptive navigation support perspective: cardinality, expressive power, and navigation.

From the *cardinality* aspect, there are essentially only two different cases: single-concept indexing, where each page is related to one and only one external model concept; and multi-concept indexing, where each page can be related to many concepts. Single-concept indexing (categorization) is simpler and more intuitive for the authors. Multi-concept indexing is more powerful, but it makes the system more complex and requires more elaborate external models. In many cases, the choice of single or multi-concept indexing is a design decision for the authors of the system. To provide some simple navigation support functionalities the authors can use or build a coarse-grain model and use single-concept indexing. To provide more elaborate adaptations, they may need a finer-grained model and apply multi-concept indexing.

The *Navigation* aspect is important when distinguishing between cases where the link between a concept and a page exists only on a conceptual level (used only by internal adaptation mechanisms of the system) from cases where each link also defines a navigation path.

Expressive power concerns the amount of information that the authors can associate with every link between a concept and a page. Of course, the most important information is the very presence of the link. This case could be called flat indexing and is used in the majority of existing systems. Still, some systems with a large hyperspace and advanced adaptation techniques may want to associate more information with every link by using roles and/or weights. Assigning a *role* to a link helps distinguish several kinds of connections between concepts and pages. For example, some systems want to distinguish whether a page provides an introduction, a core explanation or a summary of a concept. Other systems use *prerequisite* role to mark the case when the concept is not presented on a page, but instead, the page is a required prerequisite for understanding the concept [14]. A case for a more elaborate indexing with multiple roles can be found in [12]. Another way to increase the expressive power of the indexing is to specify the *weight* of the link between a concept and a page. The weight may specify, for example, the percentage of knowledge about a concept presented on this page [30; 70].

Existing AH systems suggest various ways of indexing that differ in all the aspects listed above. However, for simplicity, all this variety can be described in terms of two basic approaches that are described in the remaining part of this section. Systems using the same indexing approach have a similar hyperspace structure and share specific adaptation techniques that are based on this structure. Thus the indexing approach selected by developers to a large extent defines the navigation support functionality of the system.

Concept-Based Hyperspace. The simplest approach to organizing connections between external models and hyperspace pages is known as *concept-based hyperspace*. This approach is naturally appearing in any system that uses single-concept indexing. It is useful to distinguish simple and enhanced concept-based hyperspace. *Simple concept-based hyperspace* is used in systems that have exactly one page for every concept. With this approach, the hyperspace is built as an exact replica of the external model. Each concept of the external model is represented by exactly one node of the hyperspace, while the semantic links between the concepts constitute main paths between hyperspace nodes [17; 19; 49]. The simple concept based approach was quite popular among early educational AH systems that have their roots in the ITS field. For these systems the concept-based hyperspace was simply the easiest and the most natural way to produce a well-structured hyperspace. Currently it is rarely used in AH systems in its pure form because it requires each page of the hyperspace to be devoted to exactly one concept. It is very appropriate for developing encyclopedically structured hyperspaces such as encyclopedias [6; 62] or glossaries [14], but too restrictive for other cases.

With an *enhanced concept-based hyperspace* design approach, each concept has a corresponding “hub” page in the hyperspace. The concept hub page is connected by links to all pages categorized with this concept. For example, news articles can be classified by category and presented on a dedicated category page; Web links can be assembled under Web directory categories. The links can be typed or weighted [66; 70]. This approach is typical for adaptive e-learning systems with rich content. In this context, a variety of educational resources can be used to present different aspects of the same topic in different ways. Each page (resource) can be typed with the kind of material (video, audio, text, etc) and this typing is used for both presenting and adaptive ordering of links. With the enhanced concept-based approach users can navigate between concept pages along links that connect concepts in external models and from concept pages to the pages categorized under the concept. This approach was used for creating relatively large hyperspaces with quite straightforward structure and meaningful adaptation techniques [66; 75].

The concept-based hyperspace design approach sets strong requirements to the external model. It always requires a model with established links between concepts (preferably, several types of links) that will be used to establish hyperlinks. Another restriction is that this approach can hardly be used "post-hoc" to turn an existing traditional hypermedia system into an AH system. It has to be used from the early steps of a hypermedia system design [79]. However, this approach is quite powerful and provides excellent opportunities for adaptation. With concept-based approach, the system knows exactly the type and content of each page and the type of each link. This knowledge can be used by various adaptive navigation support techniques. Annotation is the most popular technology here. For example, ISIS-Tutor [17], ELM-ART [78], InterBook [14], INSPIRE [66] use different kinds of link annotation to show the current educational state of the concept (not known, known, well known). ISIS-Tutor, ELM-ART, and a number of other systems use annotation to show that a concept page is not ready to be learned (i.e., its prerequisite concepts are not learned yet). Hiding technology can be used to hide links to concept pages that are not relevant to the user knowledge or interests. For example, links to news categories that the user wants to ignore, links to concepts that do not belong to the current educational goal [17; 66] or with not yet learned prerequisite concepts [17; 43].

Note that the concept-based hyperspace is just one of the possible design approaches for AHS with single concept indexing. There are a few known systems, especially among early AHS [69] with single concept indexing but without concept-based navigation. The concept-based hyperspace in these systems is not formed since concepts have no external hyperspace representation and/or links between concepts and pages are purely conceptual and not used for hyperspace navigation. However, once discovered, the concept-based hyperspace approach became most popular in systems with single-concept indexing.

Page Indexing. Page indexing is the standard design approach for systems with multi-concept indexing. With this approach, the hypermedia page is indexed with several external model concepts. In other words, links are created between a page and each concept that describes the page. The simplest indexing approach is flat, content-based indexing when a concept is included in a page index if it expresses some aspect of page content. For example, the content is relevant to a specific task (a concept in a taxonomy of tasks) [76] or it presents knowledge designated by a specific domain concept [17; 47; 57]. A more general but less often used way to index the pages is to add the role for each concept in the page index (role-based indexing) as was discussed above.

Page indexing can be applied even to vector external models that have no links between concepts [32; 57]. At the same time, indexing is a very powerful mechanism, because it provides the system with knowledge about the content of its pages. With content-based indexing, the system knows quite reliably what each page is about. This knowledge can be used in multiple ways by various navigation support techniques.

Concept-Based Navigation. An interesting combination of concept-based hyperspace and page indexing known as *concept-based navigation* was introduced in InterBook [14]. This approach merges a hyperspace of multi-concept-indexed pages and a hyperspace of concepts. Each concept used to index hyperspace pages becomes a node in the hyperspace and a navigation hub. Every link between a page and the concept established during indexing becomes visible as a two-way navigational link between this page and the hub page of the concept. Thus, from any content page, users can navigate to hubs of all concepts used to index this page. Vice versa, the concept hub page provides links to all content pages indexed with this concept. This approach creates rich navigation opportunities. A user can start from a content page, move to one of the related concepts and then move to another page connected to the same concept. Concept hubs are used here as bridges for navigation to concept-related pages that have no direct hypertext links. A similar tag-based navigation approach is now popular in collaborative tagging systems, in order to navigate from one resource to another resource through *tags* [61].

The Indexing-Based Mechanism in InterBook. The InterBook system [18], the first authoring platform for Web-based adaptive hypermedia, refined the ideas of the adaptive electronic textbook introduced by ELM-ART (see section 8.3.4). A document collection in InterBook was formed by grouping several hierarchically structured textbooks into bookshelves. The books on the same shelf shared the same domain model. This domain model was used to create an overlay model of user knowledge (see Chapter 1 of this book [16])and to index each section of each book on the shelf. Connections between pages and concepts were typed: a concept served either as a *prerequisite* of a page or as an *outcome*. Following the concept-based navigation approach, each domain model concept was represented in the hyperspace as a *glossary page* that contained a brief description of the concept and links to all pages indexed by this concept. To complete concept-based navigation, every book page included a sidebar with links to all concepts used to index this page. In both contexts, the links were grouped by type, i.e., prerequisite and outcome links were not intermixed (Fig. 8.10).

InterBook offered several kinds of navigation support. The most important was link annotation, using the traffic light metaphor for adaptive navigation support in educational hypermedia (Fig. 8.10). Propagated by ELM-ART and InterBook, this metaphor has later been used in numerous adaptive educational hypermedia systems, including AST [74], KBS-HyperBook [48], and SIGUE [25]. The traffic-light annotation was produced taking into account the current model of user knowledge and the type of links between pages and concepts. A page with all outcome concepts already learned was marked with a white bullet. A page with at least some outcome concepts not learned, but with all prerequisite concepts learned was marked with a green "go" bullet. A page with at least one prerequisite concept not yet learned was marked with a red "stop" bullet. Regardless of the type of annotation, all links were functional; there was no hiding, removing or disabling. Surprisingly, the study recorded that some percentage of users most frequently chose the red link. However, it harmed their performance on tests [13]. In addition, concepts links on the concept bar were annotated with checkmarks of several difference sizes, where each size corresponded to a specific knowledge level. This feature allowed the users to see immediately which new concepts are introduced on a page and which unknown prerequisite concepts made this page hard to understand.

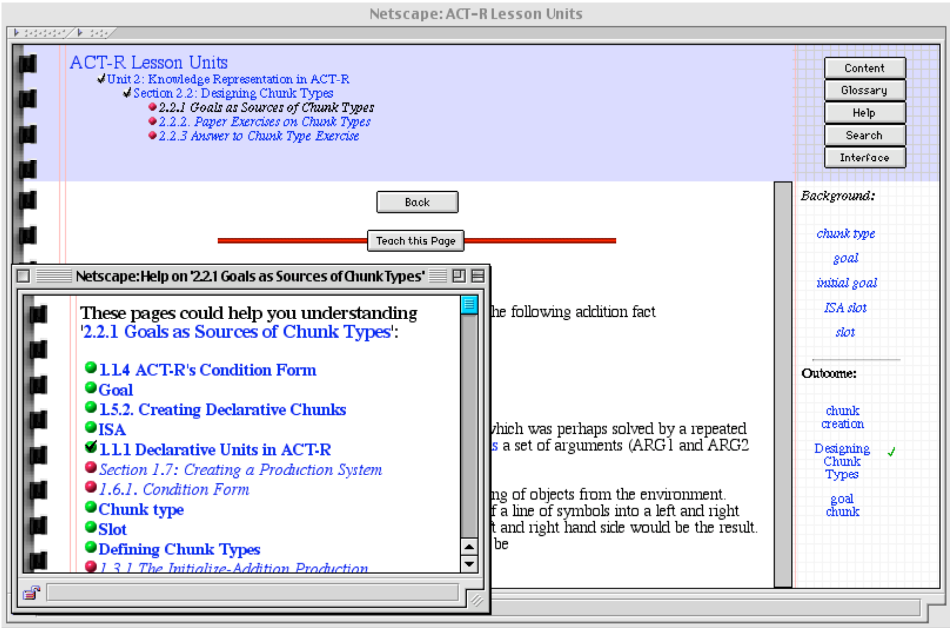


Fig. 8.10. Adaptive Navigation support in InterBook. Icons using the traffic light metaphor annotate links to book pages. Checkmarks annotate links to glossary items. By user “help” request, links to pages that can help the student to understand the current page were adaptively generated, ordered, and annotated (lower left window).

For users who have troubles selecting a link, the system offered direct guidance using a “teach me” button. The sequencing algorithm was simple: the system selected the most sequentially close green link. Finally, the system included *link generation* to answer help requests. The idea of providing help was to assemble a list of links to pages that could be useful for understanding the current not-ready-to-be-learned page. To assemble this list, the system collected all pages that might be useful for teaching the missing prerequisite concepts of the current page and ordered them adaptively, according to a polynomial “usefulness” measure. The measure took into account how many goal concepts were introduced on a page (the more, the better), how many non-goal concepts (the fewer, the better), and what the page’s current state was (green is better than red). In addition to adaptive ordering, all links were also annotated (Fig. 8.10).

Altogether, InterBook produced several kinds of navigation support using the same concept-level models for the user and the documents. A study of InterBook demonstrated that adaptive navigation support encourages non-sequential navigation and helps users who follow the system's guidance achieve a better level of knowledge [13].

8.5 Beyond Hypermedia: Adaptive Navigation Support for Virtual Environments

Adaptive navigation support techniques have demonstrated their ability to help individual users of hypermedia and Web systems. A review of adaptation techniques and mechanisms provided in this chapter could possibly serve as a collection of useful recipes for future developers of Web hypermedia systems who are interested in providing personalized assistance to users. However, a hyperspace of connected pages—which is the context of existing AH technologies—is not the only kind of "virtual space" that is available for Internet users. Even in the early days of the Internet, a lot of people were navigating in text-based virtual environments, now called MUDs and MOOs (http://www.moo.mud.org/moo-faq/) that are currently still accessible over the Web. More recently, Web-based virtual reality has become an alternative type of virtual environment for browsing and exploration on the Web. While MUD/MOO, hypermedia, and the 3D virtual environments are quite different in their nature, all these environments are targeted for user-driven navigation and exploration. As a result, in all these contexts, users can benefit from the navigation support provided by an adaptive system. We believe that the theories behind adaptive navigation support go beyond the scope of hypermedia, although a different set of technologies may be required to provide support in these different contexts.

A pioneer attempt to develop navigation support in the MOO context, using social navigation mechanisms was done by Dieberger [35] in his system Juggler. While Juggler's concept of employing history-rich environments has been explored before, Juggler suggested a unique implementation of this idea adapted to the narrative, text-based information presentation context of MOO. A number of more recent projects explored the use of navigation support in the context of Web virtual reality. For example, [51] attempted to develop virtual reality analogs to direct guidance and link annotation. A review of work in this direction is presented in Chapter 14 of this book [27].

Acknowledgments. This material is partially based upon work supported by the National Science Foundation under Grants No. 0310576 and 0447083. The author also thanks Science Foundation Ireland for its support through the E.T.S. Walton Award.

References

1. Armstrong, R., Freitag, D., Joachims, T., Mitchell, T.: WebWatcher: A learning apprentice for the World Wide Web. In: Knoblock, C., Levy, A. (eds.) Proc. of

AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments. AAAI Press (1995) 6-12
2. Barrett, R., Maglio, P.P.: Intermediaries: New places for producing and manipulating Web content. In: Ashman, H., Thistewaite, P. (eds.) Proc. of Seventh International World Wide Web Conference. Computer Networks and ISDN Systems, Vol. 30. Elsevier Science B. V. (1998) 509 - 518
3. Billsus, D., Pazzani, M.: Adaptive news access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
4. Böcker, H.D., Hohl, H., Schwab, T.: ϒπAdaptєρ - Individualizing Hypertext. In: Diaper, D. (ed.) Proc. of IFIP TC13 Third International Conference on Human-Computer Interaction. North-Holland (1990) 931-936
5. Bollen, J., Heylighen, F.: A system to restructure hypertext networks into valid user models. The New Review of Multimedia and Hypermedia 4 (1998) 189-213
6. Bontcheva, K., Wilks, Y.: Tailoring automatically generated hypertext. User Modeling and User Modeling and User Adapted Interaction 15, 1-2 (2005) 135-168
7. Boyle, C., Encarnacion, A.O.: MetaDoc: an adaptive hypertext reading system. User Modeling and User-Adapted Interaction 4, 1 (1994) 1-19
8. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction 6, 2-3 (1996) 87-129
9. Brusilovsky, P.: Adaptive hypermedia. User Modeling and User Adapted Interaction 11, 1/2 (2001) 87-110
10. Brusilovsky, P.: Adaptive navigation support in educational hypermedia: the role of student knowledge level and the case for meta-adaptation. British Journal of Educational Technology 34, 4 (2003) 487-497
11. Brusilovsky, P., Chavan, G., Farzan, R.: Social adaptive navigation support for open corpus electronic textbooks. In: De Bra, P., Nejdl, W. (eds.) Proc. of Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004). Lecture Notes in Computer Science, Vol. 3137. Springer-Verlag (2004) 24-33
12. Brusilovsky, P., Cooper, D.W.: Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System. In: Gil, Y., Leake, D.B. (eds.) Proc. of 2002 International Conference on Intelligent User Interfaces. ACM Press (2002) 23-30
13. Brusilovsky, P., Eklund, J.: A study of user-model based link annotation in educational hypermedia. Journal of Universal Computer Science 4, 4 (1998) 429-448
14. Brusilovsky, P., Eklund, J., Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware. In: Ashman, H., Thistewaite, P. (eds.) Proc. of Seventh International World Wide Web Conference. Vol. 30. Elsevier Science B. V. (1998) 291-300
15. Brusilovsky, P., Henze, N.: Open corpus adaptive educational hypermedia. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and

Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
16. Brusilovsky, P., Millán, E.: User models for adaptive hypermedia and adaptive educational systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
17. Brusilovsky, P., Pesin, L.: An intelligent learning environment for CDS/ISIS users. In: Levonen, J.J., Tukianinen, M.T. (eds.) Proc. of The interdisciplinary workshop on complex learning in computer environments (CLCE94). EIC (1994) 29-33, also available at http://cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html
18. Brusilovsky, P., Pesin, L.: Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. Journal of Computing and Information Technology 6, 1 (1998) 27-38
19. Brusilovsky, P., Pesin, L., Zyryanov, M.: Towards an adaptive hypermedia component for an intelligent learning environment. In: Bass, L.J., Gornostaev, J., Unger, C. (eds.) Proc. of 3rd International Conference on Human-Computer Interaction, EWHCI'93. Lecture Notes in Computer Science, Vol. 753. Springer-Verlag (1993) 348-358
20. Brusilovsky, P., Schwarz, E., Weber, G.: ELM-ART: An intelligent tutoring system on World Wide Web. In: Frasson, C., Gauthier, G., Lesgold, A. (eds.) Proc. of Third International Conference on Intelligent Tutoring Systems, ITS-96. Lecture Notes in Computer Science, Vol. 1086. Springer Verlag (1996) 261-269
21. Brusilovsky, P., Sosnovsky, S., Shcherbinina, O.: QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In: Nall, J., Robson, R. (eds.) Proc. of World Conference on E-Learning, E-Learn 2004. AACE (2004) 1806-1813
22. Brusilovsky, P., Sosnovsky, S., Yudelson, M.: Addictive links: The motivational value of adaptive link annotation in educational hypermedia. In: Wade, V., Ashman, H., Smyth, B. (eds.) Proc. of 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2006). Lecture Notes in Computer Science, Vol. 4018. Springer Verlag (2006) 51-60
23. Brusilovsky, P., Weber, G.: Collaborative example selection in an intelligent example-based programming environment. In: Edelson, D.C., Domeshek, E.A. (eds.) Proc. of International Conference on Learning Sciences, ICLS'96. AACE (1996) 357-362
24. Burke, R.: Hybrid Web recommender systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
25. Carmona, C., Bueno, D., Guzmán, E., Conejo, R.: SIGUE: Making Web Courses Adaptive. In: De Bra, P., Brusilovsky, P., Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002). Lecture Notes in Computer Science, Vol. 2347. Springer-Verlag (2002) 376-379

26. Cheng, R., Vassileva, J.: Adaptive Reward Mechanism for Sustainable Online Learning Community. In: Looi, C.-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proc. of 12th International Conference on Artificial Intelligence in Education, AIED'2005. IOS Press (2005) 152-159
27. Chittaro, L., Ranon, R.: Adaptive 3D Web sites. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
28. Claypool, M., Le, P., Wased, M., Brown, D.: Implicit interest indicators. In: Proc. of 6th International Conference on Intelligent User Interfaces. ACM Press (2001) 33-40
29. De Bra, P., Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia 4 (1998) 115-139
30. De Bra, P., Ruiter, J.-P.: AHA! Adaptive hypermedia for all. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet. AACE (2001) 262-268
31. De Bra, P., Smits, D., Stash, N.: The design of AHA! In: Wiil, U.K., Nürnberg, P.J., Rubart, J. (eds.) Proc. of Seventeenth ACM Conference on Hypertext and Hypermedia (Hypertext 2006). ACM Press (2006) 133
32. De Bra, P.M.E.: Teaching Hypertext and Hypermedia through the Web. Journal of Universal Computer Science 2, 12 (1996) 797-804
33. de La Passardiere, B., Dufresne, A.: Adaptive navigational tools for educational hypermedia. In: Tomek, I. (ed.) Proc. of ICCAL'92, 4-th International Conference on Computers and Learning. Lecture Notes in Computer Science, Vol. 602. Springer-Verlag (1992) 555-567
34. Debevc, M., Meyer, B., Donlagic, D., Rajko, S.: Design and Evaluation of an Adaptive Icon Toolbar. User Modeling and User-Adapted Interaction 6, 1 (1996) 1-21
35. Dieberger, A.: Supporting social navigation on the World Wide Web. International Journal of Human-Computer Interaction 46 (1997) 805-825
36. Dieberger, A., Dourish, P., Höök, K., Resnick, P., Wexelblat, A.: Social navigation: Techniques for building more usable systems. interactions 7, 6 (2000) 36-45
37. Dieberger, A., Guzdial, M.: CoWeb - experiences with collaborative Web spaces. In: Lueg, C., Fisher, D. (eds.): From Usenet to CoWebs: Interacting with Social Information Spaces. Springer-Verlag, New York (2003) 155-166
38. Dourish, P., Chalmers, M.: Running out of space: Models of information navigation. In: Cockton, G., Draper, S.W., Weir, G.R.S. (eds.) Proc. of HCI'94. (1994)
39. Dron, J., Boyne, C., Mitchell, R., Siviter, P.: CoFIND: steps towards a self-organising learning environment. In: Davies, G., Owen, C. (eds.) Proc. of WebNet'2000, World Conference of the WWW and Internet. AACE (2000) 75-80
40. Farzan, R., Brusilovsky, P.: Social navigation support in E-Learning: What are real footprints. In: Anand, S.S., Mobasher, B. (eds.) Proc. of IJCAI'05 Workshop on Intelligent Techniques for Web Personalization. (2005) 49-56,

also available at http://maya.cs.depaul.edu/~mobasher/itwp05/final/Paper7Farzan.pdf
41. Farzan, R., Brusilovsky, P.: Social navigation support through annotation-based group modeling. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) Proc. of 10th International User Modeling Conference. Lecture Notes in Artificial Intelligence, Vol. 3538. Springer Verlag (2005) 463-472
42. Farzan, R., Brusilovsky, P.: Social navigation support in a course recommendation system. In: Wade, V., Ashman, H., Smyth, B. (eds.) Proc. of 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2006). Lecture Notes in Computer Science, Vol. 4018. Springer Verlag (2006) 91-100
43. Forcheri, P., Molfino, M.T., Moretti, S., Quarati, A.: An Approach to the Development of Re-Usable and Adaptive Web Based Courses. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet. AACE (2001) 1043-1048
44. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
45. Geldof, S.: Con-textual navigation support. New Review of Multimedia and Hypermedia 4 (1998) 47-66
46. Hammond, N.: Hypermedia and learning: Who guides whom? In: Maurer, H. (ed.) Proc. of 2-nd International Conference on Computer Assisted Learning, ICCAL'89. Lecture Notes in Computer Science, Vol. 360. Springer-Verlag (1989) 167-181
47. Henze, N., Naceur, K., Nejdl, W., Wolpers, M.: Adaptive hyperbooks for constructivist teaching. Künstliche Intelligenz, 4 (1999) 26-31
48. Henze, N., Nejdl, W.: Adaptation in open corpus hypermedia. International Journal of Artificial Intelligence in Education 12, 4 (2001) 325-350
49. Hohl, H., Böcker, H.-D., Gunzenhäuser, R.: Hypadapter: An adaptive hypertext system for exploratory learning and programming. User Modeling and User-Adapted Interaction 6, 2-3 (1996) 131-156
50. Höök, K.: Evaluating the utility and usability of an adaptive hypermedia system. In: Moore, J., Edmonds, E., Puerta, A. (eds.) Proc. of 1997 International Conference on Intelligent User Interfaces. ACM (1997) 179-186
51. Hughes, S., Brusilovsky, P., Lewis, M.: Adaptive navigation support in 3D e-commerce activities. In: Ricci, F., Smyth, B. (eds.) Proc. of Workshop on Recommendation and Personalization in eCommerce at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002). (2002) 132-139, also available at http://ectrl.itc.it/home/rpec/RPEC-Papers/15-hughes.pdf
52. Kaplan, C., Fenwick, J., Chen, J.: Adaptive hypertext navigation based on user goals and context. User Modeling and User-Adapted Interaction 3, 3 (1993) 193-220
53. Kaptelinin, V.: Item recognition in menu selection: The effect of practice. In: Proc. of INTERCHI'93. (1993) 183-184

54. Kavcic, A.: Fuzzy User Modeling for Adaptation in Educational Hypermedia. IEEE Transactions on Systems, Man, and Cybernetics 34, 4 (2004) 439-449
55. Kelly, D., Tangney, B.: Matching and Mismatching Learning Characteristics with Multiple Intelligence Based Content. In: Looi, C.-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proc. of 12th International Conference on Artificial Intelligence in Education, AIED'2005. IOS Press (2005) 354-361
56. Kurhila, J., Miettinen, M., Nokelainen, P., Tirri, H.: EDUCO - A collaborative learning environment based on social navigation. In: De Bra, P., Brusilovsky, P., Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002). Lecture Notes in Computer Science, Vol. 2347. (2002) 242-252
57. Laroussi, M., Benahmed, M.: Providing an adaptive learning through the Web case of CAMELEON: Computer Aided MEdium for LEarning on Networks. In: Alvegård, C. (ed.) Proc. of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering. (1998) 411-416
58. Lieberman, H.: Letizia: An agent that assists Web browsing. In: Proc. of the Fourteenth International Joint Conference on Artificial Intelligence. (1995) 924-929
59. Lutkenhouse, T., Nelson, M.L., Bollen, J.: Distributed, real-time computation of community preferences. In: Proc. of Proceedings of the sixteenth ACM conference on Hypertext and hypermedia. ACM Press (2005) 88-97
60. Micarelli, A., Sciarrone, F., Marinilli, M.: Web document modeling. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
61. Millen, D.R., Feinberg, J., Kerr, B.: Dogear: Social bookmarking in the enterprise. In: Proc. of SIGCHI conference on Human Factors in computing systems, CHI 2006. (2006) 111 -120
62. Milosavljevic, M.: Augmenting the user's knowledge via comparison. In: Jameson, A., Paris, C., Tasso, C. (eds.) Proc. of 6th International Conference on User Modeling, UM97. SpringerWienNewYork (1997) 119-130
63. Mladenic, D.: Personal WebWatcher: Implementation and Design, Technical Report No. IJS-DP-7472, Department of Intelligent Systems, J. Stefan Institute (1996)
64. Ng, M.H., Hall, W., Maier, P., Armstrong, R.: The Application and Evaluation of Adaptive Hypermedia Techniques in Web-based Medical Education. Association for Learning Technology Journal 10, 3 (2002) 19-40
65. Olston, C., Chi, E.H.: ScentTrails: Integrating browsing and searching on the Web. ACM Transactions on Computer-Human Interaction 10, 3 (2003) 177-197
66. Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H., Magoulas, G.D.: Personalising the interaction in a Web-based Educational Hypermedia System: the case of INSPIRE. User Modeling and User Adapted Interaction 13, 3 (2003) 213-267
67. Pazzani, M., Muramatsu, J., Billsus, D.: Syskill & Webert: Identifying interesting Web sites. In: Proc. of the Thirteen National Conference on Artificial Intelligence, AAAI'96. AAAI Press /MIT Press (1996) 54-61

68. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
69. Pérez, T., Gutiérrez, J., Lopistéguy, P.: An adaptive hypermedia system. In: Greer, J. (ed.) Proc. of AI-ED'95, 7th World Conference on Artificial Intelligence in Education. AACE (1995) 351-358
70. Pilar da Silva, D., Durm, R.V., Duval, E., Olivié, H.: Concepts and documents for adaptive educational hypermedia: a model and a prototype. In: Brusilovsky, P., De Bra, P. (eds.) Proc. of Second Adaptive Hypertext and Hypermedia Workshop at the Ninth ACM International Hypertext Conference Hypertext'98. Eindhoven University of Technology (1998) 35-43, also available as http://wwwis.win.tue.nl/ah98/Pilar/Pilar.html
71. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
72. Smyth, B.: Case-base recommendation. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
73. Specht, M.: Empirical evaluation of adaptive annotation in hypermedia. In: Ottmann, T., Tomek, I. (eds.) Proc. of ED-MEDIA/ED-TELECOM'98 - 10th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications. AACE (1998) 1327-1332
74. Specht, M., Weber, G., Heitmeyer, S., Schöch, V.: AST: Adaptive WWW-Courseware for Statistics. In: Brusilovsky, P., Fink, J., Kay, J. (eds.) Proc. of Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling, UM97. (1997) 91-95, also available at http://www.contrib.andrew.cmu.edu/~plb/UM97_workshop/Specht.html
75. Steinacker, A., Seeberg, C., Rechenberger, K., Fischer, S., Steinmetz, R.: Dynamically generated tables of contents as guided tours in adaptive hypermedia systems. In: Kommers, P., Richards, G. (eds.) Proc. of ED-MEDIA/ED-TELECOM'99 - 11th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications. AACE (1999) 640-645
76. Vassileva, J.: A task-centered approach for user modeling in a hypermedia office documentation system. User Modeling and User-Adapted Interaction 6, 2-3 (1996) 185-224
77. WBT Systems: TopClass, ver. 3.0, Dublin, Ireland, WBT Systems (1999) http://www.wbtsystems.com/
78. Weber, G., Brusilovsky, P.: ELM-ART: An adaptive versatile system for Web-based instruction. International Journal of Artificial Intelligence in Education 12, 4 (2001) 351-384

79. Weber, G., Kuhl, H.-C., Weibelzahl, S.: Developing adaptive internet based courses with the authoring system NetCoach. In: Bra, P.D., Brusilovsky, P., Kobsa, A. (eds.) Proc. of Third workshop on Adaptive Hypertext and Hypermedia. (2001) 35-48, also available at http://wwwis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf
80. Wexelblat, A., Mayes, P.: Footprints: History-rich tools for information foraging. In: Proc. of ACM Conference on Human-Computer Interaction (CHI'99). (1999) 270-277
81. Yan, T.W., Jacobsen, M., Garcia-Molina, H., Dayal, U.: From user access patterns to dynamic hypertext linking. Computer Networks and ISDN Systems. (1996) 1007-1014
82. Zellweger, P.T., Chang, B.-W., Mackinlay, J.D.: Fluid links for informed and incremental link transitions. In: Grønbæk, K., Mylonas, E., Shipman III, F.M. (eds.) Proc. of Ninth ACM International Hypertext Conference (Hypertext'98). ACM Press (1998) 50-57