Adaptive Hypermedia and Hypertext Navigation

Research Review for Depth Oral Examination

Theophanis Tsandilas
fanis@cs.toronto.edu
Department of Computer Science
University of Toronto

1 Introduction

Like all men of the Library, I have travelled in my youth;
I have wandered in search of a book, perhaps the catalogue of catalogues…

The Library of Babel, Jorge Luis Borges

In his short story "The Library of Babel", Jorge Luis Borges describes an unending library consisting of an infinite number of interconnected cells. Each cell contains books whose content seems random and meaningless. Borges' story talks about book searchers who waste their lives looking for books of wisdom and die lost in the library's labyrinth. In the Greek mythology, Labyrinth was a maze built at Knossos, Crete by Daedalus to house Minautor, a half-man and half-bull creature. Theseus, the son of the king of Athens, killed Minautor and achieved to find his way out of the maze by following a thread offered to him by the Cretan princess, Ariadne.

Hypertexts are "networks of linked nodes which readers are free to navigate in a non-linear fashion" [55]. They can be thought as virtual labyrinths where users can found themselves lost if no guidance analogous to Ariadne's thread is provided. Rollason [84] identifies an analogy between the World-Wide Web (WWW), which is the most popular hypertext system, and Borges' library of Babel. The Web has been frequently described as a means where people can find any kind of information. The vast overload of this information, however, diminishes its power and usefulness. Much of the information on the Web has not been verified; the Web involves millions of documents that are not credible; Web users often do not know where to look; they have to read and assess information that may be irrelevant to their needs. In other words, as in Borges' imaginary library, a lot of information in the Web is either meaningless or inaccessible to the majority of the Web users.

Lieberman employs the metaphor of Borges' library to introduce Letizia [61], an adaptive hypermedia system which assists users in discovering interesting information in the Web. Liberman's implied argument is that adaptive interfaces could be a solution to the problem of information overload that Web users face today. Adaptive Hypermedia (AH) systems are a special type of adaptive systems in the intersection of several research disciplines such as Hypertext, User Modelling (UM), Machine Learning (ML), Information Retrieval (IR), and Human Computer Interaction (HCI). Their goal is to imitate Ariadne's role by eliminating information of low interest, providing navigation clues and recommending interesting paths of exploration. As different users have different goals and interests, AH systems adapt their assistance based on *user models* which capture information about the users.

The present review explores AH approaches whose goal is to assist navigation in hypertext environments and the Web in particular. Its main ambition is to reveal major problems that existing work in this domain has not yet addressed and suggest directions for future research. Although the review explores several user modelling and learning techniques, it mainly focuses on the usability of AH systems and the analysis of adaptation techniques that could improve the communication between AH systems and their users. Despite the progress in the development of sophisticated user models and intelligent algorithms during the last decade, adaptive systems have not succeeded in proving their usefulness. We believe that the main reason for this is that important usability issues concerning adaptive systems have been extensively disregarded. The review presents initial work that we have conducted to address some of these issues.

The rest of the document is organized as follows. In Section 2, after presenting a brief history of hypertext systems, we identify the main problems that navigation in hypertext environment and the Web in particular engages. We also discuss several factors that may affect navigation such as navigation strategies, mental models, landmarks and individual differences. We end the section with an overview of navigation tools and their main limitations. Section 3 introduces AH systems and discusses their goals and applications. Section 4 concentrates on AH systems whose main goal is to assist browsing and information discovering tasks. It overviews the most representative systems giving focus on user models and learning algorithms that these systems use. Systems that capture the navigation context are presented in Section 5. Since different approaches view context differently, we explore its definition and use from various different perspectives. Section 6 overviews representative adaptation techniques distinguishing between content adaptation and link adaptation. Problems concerning existing adaptation techniques are identified and possible solutions are proposed. Section 7 explores usability problems involved in adaptive systems. The discussion in this section concentrates on the issues of transparency and controllability. Evaluations of AH systems are investigated in Section 8. Several limitations of existing evaluations are identified. Finally, Section 9 concludes the report and discusses directions for future work.

2 Hypertext and the World Wide Web

Before introducing AH systems and investigating their challenges, we briefly overview the history of hypertext systems and discuss issues and problems associated with hypertext navigation.

2.1 A Brief History of Hypertext

2.1.1 Early Systems

Vannevar Bush (1890-1974) is usually acknowledged as the person who envisioned the first hypertext system, called Memex. Memex was described as “a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility” [16]. Bush’s ideas about Memex were developed early in 1930s, but they were not published until 1945 in the Atlantic Monthly under the title “As We May Think” [16]. Less than two decades later, in 1962, Douglas Engelbart, influenced by the ideas of Bush, started the Augment project, which resulted in the invention of half of the concepts of modern computing [74]. Part of the Augment project was NLS (oN-Line System) which had several hypertext features such as reference links between segments within files and between files.

During this time, another visionary, Ted Nelson, started developing his own vision about the creation of a universal electronic library where any substring of any document could be linked to any substring of any other document. The terms “hypertext” and “hypermedia” were coined by him. Xanadu [72], Nelson’s hypertext system, supports sophisticated techniques of versioning and copyright protection and an interactive interface where link connections between document fragments are visible. Xanadu, has been under development since then. Several other hypertext systems were developed during the next two decades. HyperCard by Apple was the most popular hypertext system in the late 1980s. HyperCard included facilities such as the back and forward buttons, the history list, and overview maps that facilitated navigation in hypertext [74].

2.1.2 *The World Wide Web*

The rapid growth of hypertext, however, occurred after the specification of the WWW by Tim Berners-Lee and his colleagues at CERN in 1990. Within only three years, the WWW and its popular browser, Mosaic, succeeded in establishing a universal hypermedia system, which has continued to grow dramatically until now. In order to get a picture of the size of the WWW, we can mention that Google indexes over 3 billion Web pages today. This number becomes more than 400 times larger if we take into account web-connected databases and dynamically generated pages [8].

2.1.3 *Open Hypermedia and the Semantic Web*

An attempt to standardize hypertext systems and provide a principled basis for comparing them was the Dexter Reference Model [39]. The Dexter model divides a hypertext system into three layers: (1) the *runtime* layer which models the presentation of the hypertext; (2) the *storage* layer which describes the basic link network structure of the hypertext; and (3) the *within-component* layer which is concerned with the contents and structure within the components of the hypertext structure. As opposed to traditional hypermedia systems where anchoring is used as the interface between the storage and the within-component layer, in open hypermedia systems [26], link information is separated from the hypertext application. Links are stored independently from the actual documents in databases named *linkbases.* This approach allows the linking between information pieces that reside in different applications such as word processors or image editors and supports sophisticated link structures such as bidirectional and many-to-many links.

The prominent Web language, HTML, does not support the separation between content and linking structures. This picture has changed after the emergence of the eXtensive Markup Language (XML). According to the specifications of the XML-based standards XPointer and XLink [99], links can be stored independently of the actual content. Different information objects can be linked together even if they reside in different applications as long as the applications support the above standards.

On the other hand, in the world of the Semantic Web [9], link structures are not pre-determined and static. They are rather dynamically inferred from semantic relationships connecting pieces of information. The goal the Semantic Web’s initiative has been to provide an infrastructure where machines would be able to process and reason about Web content. Knowledge Representation (KR) languages such as RDF, RDF Schema, and DAML+OIL [12] have been built on top of XML to provide ontological modelling and reasoning on the Web. The Semantic Web is based on an open model in the sense that semantics are stored and processed independently of the Web content. Research in the area of hypertext has used its experience from open hypermedia systems to examine how semantics should translate into link structures. The most representative work towards this direction is the COHSE project [18], which uses ontologies to define semantic relationships among concepts and then generates hypertext structures based on these relationships.

2.2 Problems Associated with Hypertext

Conklin [23] distinguishes between two main problems that users experience when navigating in hypertext systems: *disorientation* and *cognitive overhead*. According to Woods [101], disorientation, which is often described as the “lost in hyperspace” problem, occurs when “the user does not have a clear conception of relationships within the system, does not know his present location in the system relative to the display structure, and finds it difficult to decide where to look next”. Disorientation has been extensively investigated by research in hypertext. Users can loose their orientation even in small hypertext as Nielsen observed [74].

Several studies viewed and measured disorientation as a degradation in user performance. For instance, McDonald and Stevenson [65] measured disorientation in terms of the time that the users spent to complete navigational tasks and the accuracy of their answers. In their experiment, they found that the users performed better with linear text than with hypertext. On the other hand, Smith [92] proposed a set of measures to quantify "lostness" based on the number of nodes that users visit to complete a task. Otter and Johnson [77] extended these measures by considering different lostness likelihoods for different types of links. Finally, Ahuja and Webster [2] measured lostness as it is perceived by the users and distinguished it from ease of use which can also affect the performance of a user.

Cognitive overhead was defined by Conklin [23] as "the additional effort and concentration necessary to maintain several tasks or trails at one time". It can be caused when the user is given a large number of choices and needs to make decisions about which links to follow and which to abandon. Humans do not have the cognitive capability to evaluate and process all the available Web information. Cognitive overhead has been associated with the limited capacity of the human's short-term memory [40]. Zhu [104] found that the number of hyperlinks had significant effects on students' learning performance and attitude towards hypermedia systems, which was interpreted as a potential relationship between the number of links and cognitive overhead and/or disorientation. In general, it is not easy to discriminate between disorientation and cognitive overhead as disorientation can increase the cognitive overhead and both affect the performance of hypertext users.

Foss [31] grouped disorientation problems into two main categories: *embedded digression* and the *art museum* problems. Embedded digression problems arise when users navigate in highly connected networks of information. The large number of options distract them from their chosen path and cause them to lose their place in the document. Art museum problems refer to the situation where users wander through a hypertext without stopping to concentrate and assimilate the content that the document presents. As a result, the users are unable to recognize which nodes have been visited, and which parts of the documents remain to be seen.

Information overload [27, 74] and poorly designed Web sites form additional problems, which originate from the Web's rapid growth rather than the actual nature of hypertext. Taking into account the huge size of the Web, it can be expected that a large number of documents may be relevant to the interests of a user. However, it may not be clear which documents could better satisfy the user's needs or formulated differently, which navigation strategies would result in great information gains for the user [80]. The quality of the information on the Web varies greatly. Broken links, advertisements popping up, and scripts that disturb the browsing task or even crash browsers are factors that decrease the usability of the Web. These problems constitute additional sources of cognitive overhead for Web users.

2.3 Navigation Strategies

Several studies have tried to investigate and characterize user strategies in hypermedia environments. Marchionini [63] distinguishes between *analytical* and *browsing* strategies. Analytical strategies depend on careful planning, iterative query reformulations and evaluation of searching results. On the other hand, browsing strategies are heuristic and interactive, depend on recognizing relevant information and demand smaller cognitive load. Users usually employ various mixes of analytical and browsing strategies to complete their goals. According to Marchionini, people have an inclination to browse, but analytical strategies are more powerful when dealing with large collections of documents. Canter et al. [17] identify five information-seeking strategies: (1) *searching*, where users seek a specific target; (2) *browsing*, where users wander until they satisfy their interests; (3) *scanning*, where users cover a wide area shallowly; (4) *exploring*, where users survey the extent and nature of the data; and (5) *wandering*, where users navigate in an unstructured way.

Catledge and Pitkow [19] conducted an experimental study to characterize browsing strategies in the Web. Based on this study, they classified users into tree classes: *serendipitous browsers*, *general purpose* *browsers* and *searchers*. This classification was based on the average frequency of Web site visits per path length. Serendipitous browsers avoided the repetition of long navigational sequences. General purpose browsers had a 25% chance of repeating complex navigational sequences. Finally, searchers performed the same short navigational sequences infrequently, but they performed long sequences often. The study also revealed that users tend to operate in a small area within a particular site, which resembles a *spoke and hub* structure. Home pages were often used as indexes to interesting places.

2.4 Mental Models

Mental models are dynamic mental representations of the real world [63]. People construct mental models to predict and explain the operation of a target system. As Norman states [76], mental models are incomplete, unstable, unscientific and parsimonious, while humans' abilities to run them are limited. He also distinguishes between mental models and *conceptual models*, models invented by teachers, designers, scientists and engineers, which aim at being accurate, complete and consistent representations of the target system. Ideally, a system is designed and developed based on a learnable, functional and usable conceptual model which hopefully matches the mental model that the user forms.

In an information-seeking environment, mental models also apply to the knowledge domain, for example, the domain of the content in a Web site, and the underlying organizational structures, for example, the linking structure in a hypertext system. Several researchers have suggested that mental models for information spaces are analogous to mental maps of physical worlds [71]. In contrast to this assumption, Farris et al. [29] argued that users' *schemata* of hypermedia systems, i.e., their mental models, reflect not the spatial structure of the hypermedia but rather conceptual relationships of their elements. They conducted an experiment, where 40 participants were asked to draw the connection-structure of four different Web sites after exploring them. The results showed that the participants' drawings reflected semantic relationships between the individual pages and not their hypertext structure. These results signify the importance of providing navigation tools such as site maps that support navigation based on conceptual rather than spatial relationships in hypertext.

2.5 Landmarks

Landmarks are distinctive environmental features functioning as reference points [98]. They help people to remember object locations by forming distinct relationships based on visual, spatial or semantic content [62]. Landmarks have been the subject of research in the fields of urban planning, geography and psychology which explored their role in real-world navigation. According to the urban planner Lynch [62], they are one of the five key structural elements[1] which determine a user's mental map of an urban environment. Passini [78], on the other hand, identifies landmarks and paths as important way-finding elements. In addition to physical worlds, the role of landmarks is important in virtual environments, mainly when navigation is involved. In a document, elements that could act as landmarks are images, textual elements, graphics, structural forms of laying out information, fonts, etc. Hypermedia environments involve additional *structural landmarks* which are distinguishable nodes such as homepages which act as reference points in the hypertext structure. Users may highly depend on the presence of landmarks when they navigate, so disturbing these landmarks may disrupt their mental model and result in disorientation.

2.6 Individual Differences

The development of navigational strategies and mental models depends on the *personal information infrastructure* of the user, which according to Marchionini is composed of skills, knowledge, and attitudes

[1] The other four elements are *paths*, *edges*, *districts*, and *nodes*.

[63]. Attitudes include preferences, motivation, confidence, tolerance for ambiguity and uncertainty, curiosity, etc. As Marchionini states, “information problems are always embedded in a context that determines which facets of our personal information infrastructure are brought to bear in a specific situation”.

Several studies have examined the effect of individual differences such as cognitive styles and special ability to the performance of navigational tasks [20]. The most important indication of these studies is that there is correlation between spatial ability and navigation performance. More specifically, users with low spatial abilities, e.g., low spatial memory, demonstrate lower performances when interacting with systems that involve navigation. However, it seems that the use of tools such as graphical maps reduces the gap caused by differences in spatial abilities.

2.7 Navigation and Search Facilities

Hypertext browsers as well as Web sites have employed several facilities to assist the navigation tasks of hypertext users. In this section, we overview the most commonly used facilities.

2.7.1 The Backtrack Mechanism

The backtrack mechanism is the simplest and probably the most useful navigation facility. A study conducted by Catledge and Pitkow [19] found that the use of the back utility accounted for the 41% of the document requests. The most popular commercial Web browsers use the stack model to organize the history list maintained by the backtrack mechanism. However, empirical results [95] have indicated that other models based on recency might be more effective.

2.7.2 History Lists and Bookmarks

History lists and bookmarks are two other utilities employed by common browsers to assist users in returning to previously visited pages. Taucher and Greenberg [95] found that although 58% of an individual’s page accesses were revisits, the above utilities were infrequently used. In a study by Abrams et al. [1], 37% of the surveyed users declared that they did not organize bookmarks, as organizing bookmarks is a laborious and time-consuming task. Cockburn and Greenberg [22] suggested the use of visual cues such as page thumbnails and temporal information about the frequency and timing of previous visits in order to reinforce page identification and improve the usability of the history tools. Hightower et al. [41] developed a history tool which displayed interactive history maps in various scales based on Pad++, a platform for building zooming user interfaces [6]. Robertson et al. [83] proposed Data Mountain, a 3D virtual environment to organize bookmarks taking advantage of the spatial memory of users. Finally, Hunter-Gatherer by schraefel et al. [87] allowed users to maintain bookmark collections of individual page segments rather than whole pages which is the traditional approach.

2.7.3 Overview Diagrams and Site Maps

As we mentioned before, early hypertext browsers such as HyperCard supported overview diagrams. Overview diagrams visualize the structure of a hypertext and provide an alternative navigational mechanism. Although current commercial Web browsers do not support such facilities, individual Web sites provide maps that assist and direct the navigation within the site. A major challenge of overview diagrams is their scalability in large information spaces. A simple and common approach towards this issue is the use of multilevel overviews [74]. The main weakness of this approach is the lack of context as the user moves deeper to the hierarchy of the overview levels.

Fisheye views [33] address this problem by showing multiple levels of detail at the same time and combining local detail with global context. The detail in which elements of the information space are visualized is determined by the *Degree of Interest (DOI)* function whose actual formulation is application-depended. In general, fisheye-view techniques usually assume that there is a single focal point, and the value of the DOI function decreases with distance to this point. Bederson et al. [7] introduced a zooming Web browser based on fisheye views which visualized multiple Web pages at the same time at different levels of detail. Several other visualization techniques such as Cone Trees [82] and Hyperbolic Trees [58] have been proposed for displaying large hierarchical information spaces but not necessarily Web structures.

All the above techniques assume that the information space is organized into highly structured hierarchies. However, hypertext networks are often not structured as hierarchies. Several visualization techniques attempted to relax this constraint by making weaker assumptions about the structure of hypertext networks. Noik [75], for example, proposed a technique for generating fisheye views of nested graphs. Similarly to other approaches, this technique produces overviews of the information space which are hierarchically structured. The main strength of these overviews is that they show the whole information space under different levels of detail allowing for the visualization of arbitrary link structures among the visible nodes. Furnas and Zacks [34], on the other hand, introduced *multitrees*, which are basically acyclic directed graphs that consist of multiple tree structures. The main limitation of multitrees is that they do not allow diamond-like structures, i.e., two nodes cannot connect through more than one path.

Nelson, who has been a devoted opponent of hierarchies, recommended the use of cross-linked lists of nodes in order to represent arbitrary graph structures. In ZigZag [73], Nelson's alternative suggestion to structuring information, nodes can be linked together in multiple crossing dimensions. ZigZag allows users to switch dimensions and create different representations of the underlying linking structure. Nelson, however, does not distinguish between the semantics of the data, which the organization into dimensions requires, and their physical linking structure. Thus, hypertext structures cannot directly map to ZigZag structures without an interpretation of the semantics of the links. Additionally, in the lack of any empirical evaluation, no conclusion can be made about the usefulness of ZigZag's visualization as a navigational tool.

2.7.4 Search Mechanisms

The vast growth of the Web has created the need for effective techniques of searching, discovering and accessing information. Search engines like AltaVista, Yahoo and Google solved part of this problem by letting users quickly search information by typing usually a small number of keywords. The main drawback of search engines is that the results that they return are isolated from their hypertext structure and surrounding context. The initiation of a search query results in new trails which are not directly associated with the previous navigation tasks of the user. In terms of navigational strategies, search engines support analytical strategies but cannot directly assist browsing strategies.

Published studies of Web usage [19, 68, 95] have not clearly indicated how frequently search engines are used. There are, however, studies by commercial statistic services. A study by StartMarket on 2002 [94] showed that about 52% of the Web users arrived at Web sites by direct navigation or bookmarks, 41% of the users arrived by following links, and only 7-8% of the users arrived by using search engines. The above numbers do not clearly exhibit the important role of search engines. In general, users tend to discover new pages by using search engines and later navigate directly to them. As Broder et al. [11] found, a large portion of the Web's structure contains nodes which are loosely interconnected. Even nodes that are closely related may not be connected which means that search engines are often the only means of discovering new pages. Navigation by following links is usually limited within a single site, a fact that is also validated by the study of Catledge and Pitkow [19]. Unfortunately, this picture seems to be far from Bush's vision of a system that enables users to locate information by following associative trails in a way that resembles human thinking.

2.7.5 Glosses and Fluid Links

When a reader visits a hypertext page, a decision has to be made about which hyperlinks are relevant to the reader's goals and which are irrelevant. The text in the link anchors is not always adequate to describe the content of the destination page. The reader may have to follow a link in order to examine the content of the destination page and then return to the source page if this content is not interesting. A simple technique that browsers use to distinguish between visited and unvisited links is the use of a different colour for each of these two states of a link. Also, some Web sites and browsers provide *glosses* to reduce the cognitive overhead that is associated with deciding whether a link is relevant or not. A gloss provides additional information about the destination page of a link and is usually activated when the reader places the mouse over the link anchor. Zellweger et al. [103] introduced Fluid Links, a technique that uses various ways of placing glosses so that the content or layout of the source material is not obscured. The technique uses animation to achieve natural moves of the glosses between the background and the foreground. Stanyer and Procter [93] exploited the metaphor of *magic lenses* to help the user uncover information about hyperlinks. They also suggested that not only information about the content of the destination page but also information about the size and the transfer time of the page should be provided.

Although the above techniques help the users decide about the relevance of the links in a page, they still require the users to hover over the links and evaluate them by reading the attached annotations. Another weakness of these techniques is that they depend on the presence of metadata information describing the links, a requirement that rarely holds in the current Web.

3 Adaptive Hypermedia Systems

AH systems have tried to reduce problems inherent to hypermedia systems and increase their functionality. The main goal of AH systems is to provide personalized views of hypermedia responding to different goals, preferences, interests and knowledge of users. Their application is particularly useful in large hyperspaces such as the Web, where the problem of information overload is more intense. AH systems build a model of the individual user and use it to adapt the content or/and the hyper-structure of the pages in a hypermedia environment. Areas in which AH systems have been applied include educational hypermedia systems, on-line information systems, on-line help systems, information retrieval hypermedia, institutional hypermedia, and personalized views [13].

In this paper we focus on information retrieval hypermedia. This class of AH systems borrow techniques from Information Retrieval (IR) to assist the navigational tasks of hypermedia users. They have a close connection with *recommendation systems* as their main task is to identify how the various pieces of information correspond to the needs of the user and then generate personalized hypermedia views which provide hints or recommendations about the different pieces of information. We can identify three main types of assistance that such systems provide: (1) filtering, annotation or restructuring of linking information, (2) recommendation of additional information sources that may be relevant to the user's needs, and (3) enhancement of navigational tools such as bookmarks or history lists.

The goal of the first category of systems is to reduce the effects of cognitive overhead and disorientation that hypermedia users face by filtering overloading information and providing orientation cues. Several representative systems [4, 52, 69, 79] belong to this category. For instance, WebWatcher [52] suggests hyperlinks in a Web page by comparing the user's current interests with the hyperlinks' description. Syskill & Webert [79], on the other hand, rates hyperlinks by learning a user profile, which is constructed by analyzing the previous ratings of pages by the user. Finally, Personal WebWatcher [69] suggests hyperlinks based on the user's interests as inferred from the content of the documents that the user has visited in the past.

Systems that belong to the second category do not filter or rate existing hyperlinks, but they rather generate additional hyperlinks to pages whose content may be relevant to the needs of the user. Their function is similar to the function of search engines, but they do not require the user to explicitly specify his or her goals as search engines do. They rather try to automatically infer the user's goals. In addition, recommendations are integrated into the current task of the user and the hypertext structure of the visited pages. A representative example of this category of systems is Letizia [61]. Letizia is an agent that searches the space of linked documents and discovers documents that match user interests. The most relevant documents are suggested to the user. User interests are approximated by using not only the history of visited documents but also browsing times and user behaviour such as saving documents.

The augmentation of navigational tools with adaptive features has been investigated by a few only systems. Their main goal is to make navigational facilities such as bookmarks more usable by automating some of their functionality. For instance, PowerBookmarks [60] is an adaptive system which supports automatic collection of bookmarks by analyzing the frequency of page visits and the hyperlink connections between visited pages. It also supports facilities for sophisticated querying and navigation of bookmarks. Tsandilas and schraefel [97], on the other hand, proposed a system in which history lists are organized and filtered with respect to explicitly determined user interests.

Recommendation systems can be classified into two major categories: *content-based* recommendation systems, and *collaborative* recommendation systems. Content-based recommendation is based on the content of the documents under consideration and on how similar to the interests of the user this content is. On the other hand, collaborative recommendation is based on ratings of other users whose profile is similar to the profile of the user. Collaborative recommendation is mostly applicable in specific domains such as music or cinema, where users can express their utilities over items like songs, movies, etc., and user groups with similar preferences can be identified. Its main advantage over the content-based approach is that even documents whose content does not relate to the content of pages that the user has visited in the past may be recommended to the user. However, in large and heterogeneous information spaces like the Web, the existence of user communities with common information needs is questionable. As McKenzie and Cockburn found in their study [68], in a total of 16290 distinct pages accessed by a fairly homogeneous group of 17 subjects, only a 10% of the accessed pages were visited by more than one user. Collaborative recommendation is out of the scope of our interests, so the rest of the document concentrates on systems that provide content-based recommendation.

4 User Models and Learning

In this section, we examine how user models are represented, captured and used by representative content-based recommendation systems in hypermedia environments. We overview how the content of pages is represented and how information about the user is modeled and learned by such systems. The systems that we present here build persistent user profiles that represent global user interests. In Section 5, we investigate more sophisticated techniques which try to capture the context of the user's navigation and describe local user interests.

4.1 Representation of Documents

Most content-based recommendation techniques use vectors of terms, also called *feature vectors*, to represent documents. Each element in the feature vector of a document d represents a particular word w in the English language, also called *feature*, and is assigned a weight, which represents the relative weight of the corresponding word in d. Weights are computed using the *term-frequency, inverse-document-frequency* (TFIDF) heuristic [85], which is defined by the following expression:

$$TFIDF(w,d) = TF(w,d) \cdot \log(\frac{n}{DF(w)})$$

The *term frequency TF(w,d)* term counts the number of times that word *w* appears in document *d*. The *document-frequency DF(w)* term counts the number of documents that contain word *w*, while *n* represents the total number of documents. The intuition behind the TFIDF heuristic is that high weights should be assigned to words that occur frequently in the document but are not common in many documents, i.e., words that are more discriminating. There are also approaches that suggest the use of sequences of words (n-grams) instead of single words as features [70].

Not all the systems use the above technique to compute weights. In Syskill & Webert, each feature has a Boolean value that indicates whether a particular word is absent or present at least once in a Web page. On the other hand, Balabanovic and Shoham [4] use a more sophisticated TFIDF scheme, which normalizes for document length.

As the number of words in a set of documents is usually large, various feature-selection techniques are used to reduce the dimensionality of the feature vectors. Using *stop-lists* of common English words, e.g., *the*, *a*, *of*, *with*, pruning very frequent words, and stemming are the most popular approaches. Stemming consists of using a single feature to represent words of the same root, e.g., *write* and *writing* are both represented by the feature *writ-*. Most sophisticated techniques use *information gain* as a measure to select features [69, 79]. These approaches select words that make better distinction among the different classes of documents, i.e., between interesting and non-interesting documents. Sebastiani [88] presents an extensive overview on techniques for dimensionality reduction of feature vectors, which are used for text categorization, but they are not necessarily exploited by existing recommendation systems. Experiments indicate that in some cases, the best classification results can be obtained by using only a small percentage, up to 10% of carefully selected features [70].

The above techniques do not take into consideration any information about the syntax and the semantics of the words in the text. Approaches that used syntactic information did not give encouraging results, so the question whether syntax can help is still under investigation [88]. Green [37] uses lexical chains to represent semantic relationships between words that occur throughout a text. Each lexical chain is a set of related words that captures a portion of the cohesive structure of the text. This representation is used to build links between paragraphs of the same text or links between different texts.

4.2 User Profiles and User Feedback

Recommendations in content-based systems are based on a user profile which usually reflects the interests of a particular user. In WebWatcher, the user's profile simply consists of the set of keywords that the user explicitly declares as his or her interests. In Personal WebWatcher a user profile is a function $U{:}h \rightarrow \{pos, neg\}$ which assigns a Boolean value *pos* or *neg*, representing interest or lack of interest, respectively, to each hyperlink *h*. This profile is learned offline based on the hyperlinks that the user has followed in the past. Alternatively, *h* stands for document, and in this case, the profile is learned based on the documents that the user has visited in the past. Syskill & Webert uses a similar representation for a user profile, with the difference that user interests are represented by real values between 0 and 1 rather than Boolean values. Such a value denotes the confidence that the user is interested in a particular document. Finally, Balabanovic and Shoham [4] represent user profiles as feature vectors similar to the ones used to represent documents. Each page $\vec{V}_i$ receives an evaluation e_i by the user. The user's profile $\vec{M}_t$ is then updated as follows:

$$\vec{M}_t \longleftarrow \vec{M}_{t-1} + \sum_{i=1}^{p} e_i \vec{V}_i$$

The above expression implies that user interests are assumed to be independent. For instance, if a user is interested in pages about Canada as well as in pages about recipes, the user is assumed to be also interested in Canadian recipes, which may not be correct.

In recommendation systems, user feedback can be either explicit [4, 52, 79] or implicit [61, 69]. In the IR community, explicit feedback is known as *relevance feedback*. Users can provide explicit feedback by either directly specifying their interests [52] or by rating documents [4, 79]. The weakness of the first approach is that interests cannot always be described by a set of keywords. The second approach, on the other hand, requires users to rate pages that they visit, which is something they are not always willing to do. Moreover, a user profile can be learned only if there exists an adequate number of documents that have not been rated by the user. This weakness also applies to implicit feedback where knowledge about the user's interests is obtained by examining the past exploration behaviour of the user. The main problem with implicit feedback, however, is that it is not always clear how past user behaviour should be interpreted. The fact, for example, that a user has visited a page does not necessarily imply that the user is interested in that particular page. Letizia uses various heuristics to derive user interest or disinterest. For instance, it counts the time that a user spends on a page or assesses situations in which the user returns to previously visited documents or consistently passes over specific links. This approach is extended by Goecks and Shavlik [36] to observe additional user behaviour such as scrolling and mouse activity.

4.3 Learning

In this subsection, we overview learning techniques that are useful in learning profiles of users who explore hypertexts.

4.3.1 Text Classification

Text classification is one of the common techniques used for content-based recommendation. It reduces the problem to identifying the class to which a document or link belongs to. A class may represent a thematic category or a degree of user interest, e.g., class of interesting documents and class of non-interesting documents. Classification relies on the availability of an initial corpus of pre-classified documents, called *training set*. In the case of recommendation systems, the training set derives from the user feedback and grows as the user provides more feedback.

Decision trees, Bayesian classifiers, regression methods, cosine similarity, neural networks, the Rochhio method, k-Nearest Neighbour, Support Vector Machines, and classifier committees are classification techniques that are used in the community of text categorization [70, 88]. There are indications that regression methods, k-Nearest Neighbour, Support Vector Machines, and classifier committees have the top performance, while Naïve Bayes classifiers and the Rocchio method look the worst classifiers [88]. However, classifiers behave differently on different testing sets, and different experiments have produced controversy results, so final conclusions cannot be easily derived.

Here, we briefly describe the Naïve Bayes classifier, which is one of the most popular and widely tested classifiers. Given a document *d* with *n* words $(w_1, w_2,\ldots, w_n)$ and a set of document classes $C=(c_1,, c_2,\ldots, c_m)$, the probability that *d* belongs to c_j can be computed by applying the Bayes theorem as follows:

$$\Pr(c_j \mid d) = \frac{\Pr(c_j)\Pr(d \mid c_j)}{\Pr(d)} \approx \frac{\Pr(c_j)\prod_{i=1}^{n}\Pr(w_i \mid c_j)}{\Pr(d)}.$$

The above equation makes the assumption that the occurrence of a given word in a document is independent of all other words in the document, given the class. Although this assumption does not hold in real text documents, in practice, Naïve Bayes classifiers usually perform well [59]. In order to decide on which class a document belongs to, we simply calculate the expression:

$$\arg\max_{c_j \in C} \Pr(c_j) \prod_{i=1}^{n} \Pr(w_i \mid c_j).$$

One common technique to calculate the word probability estimates $Pr(w_i|c_j)$ is to use Laplace estimates:

$$\Pr(w_i \mid c_j) = \frac{N(w_i, c_j) + 1}{N(c_j) + T}$$

where $N(w_i,,c_j)$ is the number of times w_i appears in the training set for class c_j, $N(c_j)$ is the total number of words in the training set for class c_j, and T is the total number of unique words in the corpus. This technique is used to avoid the derivation of probabilities with zero values in the case of infrequently encountered words.

Hypertext documents exhibits several characteristics that are not exploited by standard text classification techniques. Hyperlinks, HTML tags, category labels, and metadata extracted from related Web sites provide rich information for classifying hypertext documents. Yang et al. [102] define five types of regularities of hypertext that can be exploited when classifying documents. The idea behind these regularities is that documents that are linked together may exhibit particular classification patterns. In the best case, each document is linked only to documents of the same class. In the worst case, no hypertext regularity exists, so additional hypertext information does not help and in some cases hurts classification performance. Metadata regularities in hypertext documents also exist. For instance, text within tags such as `META`, `ALT` and `TITLE` can help the selection of features. Hypertext classification techniques have not been adopted by known recommendation systems yet, mainly because they emerged only recently and are still under investigation.

4.3.2 Evaluation Measures

Experimental evaluations usually measure the effectiveness of a learning technique rather than its efficiency in terms of learning time. The classic IR measures of *precision* (π) and *recall* (ρ) are usually used to evaluate the effectiveness of a text classification technique. Precision π_i with respect to a class c_i is defined as the probability that if a random document d is classified under c_i, this decision is correct. Recall ρ_i with respect to a class c_i is defined as the probability that if a random document d ought to be classified under c_i, this decision is taken. Estimates of these two measures can be obtained as:

$$\pi_i = \frac{TP_i}{TP_i + FP_i}, \quad \rho_i = \frac{TP_i}{TP_i + FN_i}$$

where FP_i *(False Positives)* is the number of test documents incorrectly classified under c_i, TP_i *(True Positives)* is the number of test documents correctly classified under c_i, and FN_i *(False Negatives)* is the number of test documents incorrectly classified under another class. Estimates for total precision and recall can be obtained by *micro-averaging*[2] [88]:

$$\pi = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)}, \qquad \rho = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)}$$

There is usually a trade-off between precision and recall, so the two measures should be used in combination in order to derive meaningful conclusions about the effectiveness of a technique.

Another commonly used measure is *accuracy* which is estimated as follows:

$$a = \frac{TP + TN}{TP + TN + FP + FN}$$

[2] *Sometimes, macro-averaging is used. In this case, precision and recall are estimated by calculating the average of all* π_i and ρ_i, respectively.

where $TN = \sum_{i=1}^{|C|} TN_i$, and TN_i *(True Negatives)* is the number of test documents that are correctly not classified under class c_i.

The use of the above measures can be extended to evaluating the effectiveness of recommendation techniques. For instance, the precision of a recommendation technique can be defined as the probability that given a random document *d* which is recommended to the user, this document is relevant to the interests of the user. The importance of each measure varies among different applications. As an example, in case a large number of relevant documents exist, precision is more important, since it may not be appropriate to recommend all the relevant documents to the user. On the other hand, when only a few relevant documents exist, recall is more important, since the goal now is to present all the relevant documents to the user even with the cost of recommending some irrelevant ones.

4.3.3 Learning in Content-Based Recommendation Systems

Text classification has been used by several content-based recommendation systems. In Personal WebWatcher, both a Naïve Bayes and a k-Nearest Neighbour classifier have been applied [69] to classify documents or hyperlinks into a binary *(pos, neg)* set of classes, which represent presence or absence of user interest, respectively. The k-Nearest Neighbour classifier performed better than the Naïve Bayes classifier in most experiments in terms of accuracy and precision, although the difference was not significant. Pazzani et al. [79] investigated several classification algorithms including a Naïve Bayes classifier, the k-Nearest Neighbour classifier, and neural networks. In their experiments, the Naïve Bayes classifier had the best accuracy in most cases.

Joachims et al. [52] and Balabanovic and Shoham [4], on the other hand, estimate the similarity of a document with a user profile by calculating the cosine between the corresponding vectors:

$$\cos(\varpi) = \frac{\vec{V} \cdot \vec{M}}{|\vec{V}||\vec{M}|}$$

where $\vec{V}$ is the document vector, $\vec{M}$ is the vector representing the user profile, and ω is the angle between the two vectors.

The former work improved the accuracy of the recommendation by combining the above technique with a technique based on reinforcement learning. According to this approach, a user is considered as an agent who explores Web pages, which correspond to states, by following hyperlinks, which correspond to actions. Rewards are assigned to each state reflecting the user’s interest on the corresponding page. Sate rewards are estimated by using the cosine similarity heuristic. At each state *s*, the agent has to decide on the next action *a*. The goodness of an action is represented by a function *Q(s, a)*, which can iteratively be approximated as follows:

$$Q_n(s,a) = R(s') + \gamma \max_{a' \in actions\_in\_s'} [Q_n(s',a')]$$

where *s’* is the state resulting from performing action *a* in state *s*, *R(s’)* is the reward in state *s’* and γ is a discount factor that determines how severely rewards of future states should be taken into account. If $\gamma=0$, then the technique reduces to the pure cosine similarity approach. If $0<\gamma<1$, the goodness of an action depends not only on the reward of the next state but also on the reward of the states in the neighbourhood of this state. In other words, interesting pages that are also surrounded by other interested pages are the ones of greatest interest to the user.

5 Navigation Context

The approaches that were investigated in Section 4 attempt to derive a user profile that reflects persistent user interests. A single user, however, may have several interests and goals which can be demonstrated even during a single browsing session. A visit of a single page may initiate a brief or long shift of the user's navigation goals. Assume, for instance, that looking for publications on Adaptive Hypermedia, a user visits the homepage of a researcher. A paragraph about Toronto that appears in this homepage initiates a new exploration process in which the user searches information about Toronto. Later, the user returns to the first exploration task. Each task involves different information needs so the question that arises is how an adaptive system could support the co-presence and alternation of multiple navigational goals.

Several approaches propose techniques that capture the *context* in which the navigation takes place and assist the user based on this context. The term "context" has several meanings and interpretations. In this section, we try to identify the notions of context that relate to hypertext navigation and examine how AH systems employ them.

5.1 Defining Context

Before constraining the notion of context within the frame of hypermedia and recommendation systems, we briefly investigate how it is defined or used in other disciplines.

5.1.1 Context in the Literature

In general, context is part of the explanation of how our mental contents are used in common sense reasoning, natural language understanding, learning, problem solving etc.[35] It has been the subject of study in many disciplines such as philosophy, linguistics, psychology, theory of communication, cognitive science, and artificial intelligence. McCarthy [64] proposed the formalization of contexts and their representation as abstract mathematical entities. Under his perspective, contexts are not defined and, as rich objects, they cannot be always completely described. A "side-effect" of the context-formalization approach is that a context is always described within another context, which means that there is no an ultimate outer context representing the "absolute truth". Giunchiglia and Bouquet [35] state that "context is a partial and approximate theory of the world which encodes some agent's perspective", supporting the subjective notion of context.

Hirst [43] rejects the formalization of context as something where propositions are true in, as McCarty proposes. He rather sees context as a psychological and social construct, which is connected with interpretation and belief and is constructed dynamically through communication and social interaction. There is a speaker who constructs the elements of the context to be used in the conversation, and an interpreter who chooses which context elements to attend and interpret. Cooperative conversation results from the harmonization between the contexts of the speaker and the interpreter.

5.1.2 Context in Recommendations Systems and Adaptive Hypermedia

The term "context" is widely used in the literature of recommendation and adaptive hypermedia systems without always being explicitly defined. Kaplan et al. [53] state that context includes any number of factors that might be relevant to anticipating the user's information needs such as the time and the place of the user's interaction. In their system, context consists of user goals and user preferences for specific topics. El-Beltagy et al. [28] define context as the combination of both the user interests and the document that is currently requested by the user. Other works refer to context as the user's browsing context [5, 42, 57], the context of a task [15], or a document's spatial context as viewed from a user's perspective [3].

It can be concluded that these approaches adopt the notion of subjective contexts, which cannot be completely described. According to the needs of the particular application, different elements of context are chosen and represented by each system. Context, however, is not considered as a container of subjective propositions or subjective views of the world. Following the linguistic notion of context as presented by Hirst, it rather forms elements of interpretation of utterances expressed by the user in terms of interactions with the system. It is constructed dynamically through communication and negotiation between the user and the system, as it also depends on the system's interpretations and responses, which constrain or alter the user's mental state.

We should note, though, that the resources available to a computer system when interpreting the actions of a user are limited. Therefore it is not clear how computers could derive rich and valid contexts which have not been explicitly articulated [30]. Another problem involved in the case of adaptive systems is that users may not be aware of the interpretation model of the system, which may prevent them from "harmonizing" their context with the interpretation context. Such an interaction can be parallelized with the situation where a speaker has no information about the hearer, so he or she cannot decide on what form of communication to use in order to avoid misinterpretation. Under an HCI perspective, this problem can be viewed as the gap between the conceptual model of a system and the mental model of the user and constitutes the main subject of criticism against adaptive interfaces [90]. In Section 7, we discuss this problem in more detail.

5.2 Representing and Learning Context

There are several *context-aware* approaches in the area of AH systems, which view, represent and capture context differently. We group the different approaches according to how context is represented and learned. We should note that the user profiles used by the content-based recommendation systems that we presented in Section 4 capture a part of the browsing context in terms of global user interests. However, here, we are more interested in local contexts, which evolve and change as user interests and themes change.

5.2.1 Contexts as Browsing Histories

Representing context as the vector representation of the document that the user currently reads is the simplest approach to context. Margin Notes [81] adopt this approach to recommend documents to the user. The same approach is used in Watson [15], with the difference that Watson is able to extract terms from a variety of applications, e.g. Microsoft Word, and pre-clusters relevant documents before recommending them to the user. In addition to the current page of focus, Kushmerick et al. [57] capture information from the *referrer page*, i.e., the page from which the user came to the current page. In their prototype they use only the information that appears in the URL of the referrer page, which is assumed to represent the current browsing context.

The whole browsing path is captured in the system of Hirashima et al. [42]. According to this approach, contexts are also represented as weighted terms. Weights are updated incrementally so that high weights are assigned to terms that have appeared frequently in recently browsed documents. A parameter r with values between 0 and 1 determines the sensitivity of the context to the browsing history. When $r=0$ only the current page determines the browsing context. When $r=1$ all the visited nodes equally contribute to the calculation of the context. The main drawback of this technique is that it suffers from a trade-off between remembering past user interests and handling sudden changes of browsing contexts.

A more sophisticated approach is introduced by WordSieve [5], a contextual model consisting of a network with three levels. Each level consists of a set of words accompanied by one or two real numbers. The documents that the user visits pass through the network as streams of words. The first level of the network captures the most frequent words of the stream. A weight that accompanies each word, called *excitement*, represents the relative frequency of occurrence of the corresponding word in recent texts. Discriminator words such as “and” and “or” are also trapped by this level. The second level has only access to the first level and identifies words that tend to occur together in sequences of document accesses. Words that tended to occur frequently in the past, not necessarily recently, get high excitements. Finally, the third level has only access to the second level and learns high excitements for words that occur infrequently for periods of time. This level eliminates discriminator words. The current browsing context is calculated by multiplying the excitement values of words of all the three levels. In this way, context is represented by words that are part of both the current focus of the user and the persistent user interests.

5.2.2 Contexts as Clusters of User Interests

Several approaches [3, 28, 66] represent contexts as clusters of documents that have been visited by the user in the past. Each cluster represents a different theme of user interests. As the user’s interests change, different clusters are used to represent the current browsing context. All the three above works use TF-IDF in conjunction with a cosine similarity function to determine the clusters. The actual representation of a context consists of the vector representation of the cluster’s *centroid*. A cluster centroid is calculated by averaging the vectors of the documents in a given cluster.

El-Beltagy et al. [28] use the document contexts to extract hyperlinks and construct linkbases grouped by context which associate text anchors to page URLs. When a new page is requested by the user, it is compared to the existing contexts. If a matching context is identified, phrases in the text of the page that match anchors in the linkbase are linked to the corresponding URLs with respect to the underline context. Bailey et al. [3] extend this work by observing that a particular page can involve more than one context, e.g., a page about Canadian hockey could belong to the context “Canada” or the context “hockey”. According to their approach, the decision about which context to use depends on previous navigation contexts. For instance, if the user visited the above page coming from pages with information about Toronto, the context “Canada” is chosen.

5.2.3 Contexts as Semantic Networks

HYPERFLEX [53] defines context as a set of associative matrices representing semantic relationships between context elements. In the system’s prototype, two associative matrixes were included: one matrix containing associations between topics, and one matrix containing associations between topics and goals. Associations are weighted by integer values denoting the strength of the corresponding relationships. The assumption that the approach makes is that each document is known to belong to a particular topic. A document is considered relevant to the current context and is recommended to the user if its topic has strong relationships with the current topic and the current goal in the corresponding associative matrices. The system stores different contexts, i.e., pairs of associative matrices, which are considered as different user profiles. Users are allowed to chose among the existing profiles, combine more than one profile or build their own profiles. A profile can be built by user feedback on the system recommendations. More precisely, the user can change the ordering in the list of recommended topics updating in this way the corresponding weights in the associative matrices. An automatic technique is also provided which learns weights in a goals-topics associative matrix by observing the time the user spends on different topics given a goal.

Related to HYPERFLEX is work in the IR community which suggests the use of Bayesian networks to represent and infer relevance of documents or relationships among topics [32, 100]. These techniques remedy the naïve assumption of HYPERFLEX that document topics are known beforehand by making use of the terms that appear in the documents.

6 Adaptation Techniques

Brusilovsky [13] distinguishes between two types of adaptation that are applied by AH systems: (1) adaptive presentation, and (2) adaptive navigation support. The full taxonomy of adaptation techniques as identified by Brusilovsky is presented in Figure 1.

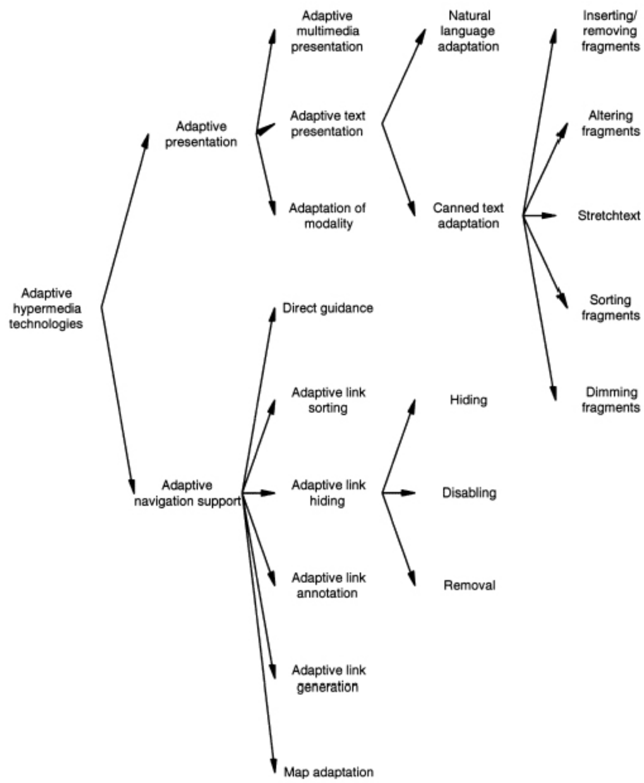


Figure 1. Brusilovsky’s Taxonomy of Adaptation Techniques

6.1 Adaptive Presentation

The goal of adaptive presentation techniques is to adapt the content of a page according to the knowledge, goals and other characteristics of the user. As the content of a page can contain both text and multimedia objects such as images, Brusilovsky’s taxonomy distinguishes between adaptive text presentation and adaptive multimedia presentation. Some approaches suggest the use and alternation of different media such as text, video, animation and sound, according to the user’s characteristics and goals, i.e., the modality of the hypermedia content is adapted [14]. Adaptive presentation is mainly used to reduce the information overload within hypertext pages, eliminate scrolling and subsequently, relax the problems of lostness and cognitive overhead. The most popular method of content adaptation is hiding from the user part of the information that is not relevant to the user’s knowledge or interests. The main disadvantage of hiding or altering information in the page’s content is that users are deprived from information that might be useful as adaptation algorithms cannot perfectly determine the exact needs of users. A technique that addresses this problem is *stretchtext*. Stretchtext is a kind of hypertext where text can be collapsed or expanded by clicking on “hot words”.

MetaDoc [10] applied stretchtext to adapt the content of pages according to the expertise of the users. Experts were provided with fewer details about various concepts. However, they could expand text with more detailed descriptions by clicking on “hot words” that appeared in the text. Similarly, novice users could collapse detailed text and reduce the information overload in the page. Stretchtext was also employed by the PUSH [47] system to adapt the presentation of learning material about a software method according to different user tasks.

In contrast to the above techniques, *sorting* or *dimming fragments* can help the user to identify useful information without hiding non-relevant information. Sorting fragments can minimize the scrolling effect as the most relevant page fragments are presented at the top of the page. The main disadvantage of this approach is that reordering fragments within a page can disturb the natural flow of information and the resulting document views may not be comprehensible. In addition, the mental model of the hypertext reader may not be able to interpret the reordering mechanism and the relationship between the different page fragments. Dimming was introduced by Hothi et al. [50]. According to this approach, document fragments that are not relevant to the user’s goals are shaded. The main drawback of dimming is that it does not reduce the size of the adapted page so the scrolling effect is not addressed. Additionally, although shaded, irrelevant information can easily gain the attention of a user; in other words, the problem of information overload is not adequately tackled by dimming.

6.1.1 Adaptive Presentation Supporting Focus and Context

Tsandilas and schraefel [96] suggested that content adaptation should be seen as a process of moving the focus within hypertext pages while always keeping out-of-focus information as context which is always visible. They introduced a fisheye-like technique to adapt the content of Web pages, which can be added to Brusilovsky’s taxonomy as a sixth technique of canned text adaptation. According to this approach, adaptation is performed by adjusting the level of zooming of page fragments. This is achieved by modifying the font size of the text as well as the size of other visible page elements such as images. More precisely, the content of a page is considered as a collection of segments, e.g., paragraphs or sections. The zooming level of the page elements in each segment s_j is determined by a function $DOI(s_j)$, which is defined as the relevance between the content of the segment and the current interests of the user. Segments that are relevant to the interests of the user appear in normal sizes while less relevant or irrelevant segments are minimized.

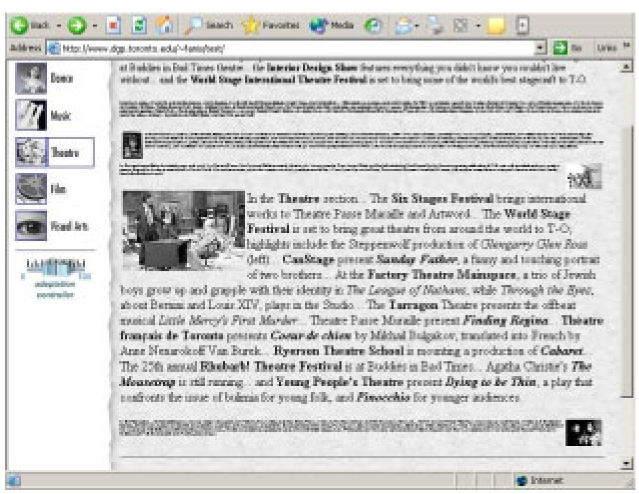


Figure 2. Fisheye-like content adaptation

In contrast to the original conception of fisheye views where proximity is measured in terms of geometric distance, the above definition of the DOI function views proximity as the semantic distance between the content of the different segments in a page. Furthermore, the focal point is determined by the focus of the user’s interests rather than the user’s current focus of attention. The approach also distinguishes between global and local changes in the user’s focus. Global focus is determined by the DOI function. The user, however, can change the local focus within a page and zoom in minimized segments by double-clicking on them. Zoom-ins and zoom-outs are animated so that the user’s view changes smoothly. We can identify several advantages of the zooming technique over stretchtext. First, it provides immediate feedback about the quantity and layout of the page fragments that appear in context. Second, it can be easily automated as it does not require the selection of representative “hot words” to represent hidden information. Finally, it can represent and visualize multiple states of adaptation. This means that it can represent multiple variations of relevance between page fragments and interests or capture the uncertainty of the user’s interests. Figure 2 presents the snapshot of a page which has been adapted using this technique.

The zooming adaptation technique was influenced by other fisheye-view approaches [7, 38, 45] which applied zooming to the content of single documents. Each of these approaches had different goals and was applied in different domains. Greenberg et al. [38] used fisheye views of text viewers to support group awareness when multiple people work within a single document. Holmquist’s Flip Browser [45] allowed easy transitions between focus and context of different parts of a document by visualizing the thumbnails of the pages that are out of focus. Finally, Bederson et al. [7] developed the *Multi-Scale Markup Language (MSML)*, a mark-up language implemented using the HTML `<Meta>` tag to enable multiple levels of zooming within Web pages. The goal of this approach was to produce interactive Web pages which can be zoomed-in and zoomed-out by the user.

A serious problem that content adaptation involves is that it may destroy landmarks within a page on which users base their reading or navigational tasks. Event content which is irrelevant to a user’s goals may serve as a landmark, so removing it from a page could disrupt the task of the user. An advantage of fisheye views over other visualization techniques is that they preserve landmarks of the context information. As shown in Figure 2, distinct structural elements of the page such as pictures, layout and number of paragraphs are maintained; they are, though, distorted. Two experimental studies conducted by Skopik and Gutwin [91] on distortion-based fisheye views of graphs revealed that distortion may not injury the spatial memory of the users as long as the users can identify and trust landmarks in the visualized space. Future research on AH systems should recognize the importance of landmarks and investigate their role in the effectiveness of different adaptation techniques.

6.1.2 Evaluation of Content Adaptation Techniques

Adaptive presentation techniques cannot be easily separated from the systems in which they have been used. Most AH systems have been based on a single adaptation technique so there are no general conclusions about how different content adaptation techniques compare in practice. For example, the evaluations of MetaDoc [10] and PUSH [47] showed that the adaptive versions of the systems improved the users' performance in several information tasks, but they did not explain whether it was the particular adaptation technique, i.e., stretchtext, or it was the efficiency of the adaptation mechanism that resulted in the observed improvements in the performance of the users.

Tsandilas and schraefel [96] conducted a pilot study to compare the zooming adaptation technique against stretchtext adaptation by measuring the performance of six subjects in several information locating and information gathering tasks in three different pages. These tasks involved searching information that was either in focus or in context. The study showed an advantage of the zooming technique in the case of the smaller pages (6-8 paragraphs long), while stretchtext performed better in the case of the large page (75 paragraphs long). The latter result can be explained by the additional scrolling that the zooming technique involves, which becomes more intense in the case of large pages. In terms of user preferences, the stretchtext technique was more preferable for locating information in focus, while the zooming technique was more preferable for locating information in context. We should also note, however, that the small number of subjects that participated in the study does not allow for general conclusions. In addition, the study did not measure the effect of several adaptation parameters, for example, font sizes, in the users' performance. Evaluating and comparing the different content adaptation techniques in separation from the underlying adaptation algorithms is still an open issue that needs further investigation by future research.

6.2 Adaptive Navigation Support

Brusilovsky's taxonomy specifies six main types of adaptive navigation support: direct guidance, adaptive link sorting, adaptive link hiding, adaptive link annotation, adaptive link generation, and map adaptation. Different combinations of the above techniques have been employed by AH systems to direct the navigational tasks of users or assist them in discovering interesting information. WebWatcher [52] and Personal WebWatcher [69] apply direct guidance by highlighting a small number of links as the most relevant to a user's interests. Syskill & Webert [79] visualizes multiple degrees of suggestions by annotating links with an icon indicating likelihood of user preference together with the estimated probability that the user would like the target page.

HYPADAPTER [44] applies different font sizes together with link sorting to adapt links. Multiple font sizes are also used by Tsandilas and schraefel [97] to represent multiple degrees of relevance between links and user interests. According to this approach, links relevant to the interests of the user are visualized with large fonts, while less relevant links are presented with smaller fonts. The main advantage of the technique is that font sizes take continuous values allowing for smooth transitions of the user's view as interests are controlled by the user. Adjusting the colour of the links is another adaptation technique that the above work applied. This technique adapts the hue of the link's colour according to the interests of the user. Relevant, i.e., "hot", links are coloured red, irrelevant links are coloured blue, while hues between red and blue are used to represent other degrees of relevance. The main disadvantage of adapting the font size or the colour of a link is that such changes may disturb the layout of the displayed information. Colours or font sizes may have been intentionally selected by the author of a page and their modification may affect the structural or semantic coherence of the page.

The above approaches adapt existing links in the page with the goal to reduce the information overload, and guide the user's navigation. Link generation techniques, on the other hand, create additional links which initially, did not appear in the page. We can distinguish between two different types of link generation: (1) generation of lists of recommended links, and (2) link augmentation. The first type of link generation is usually combined with link sorting, so that the most relevant links appear first in the list of recommendations [42, 53]. In addition to recommending links, Letizia [61] annotates links with an explanation about why they are recommended. On the other hand, instead of presenting long lists of links, Watson [15] clusters the recommendations and presents them as different content categories.

Link augmentation involves direct link insertions into the body of a document [3, 28]. More specifically, words or phrases in the document's text are selected as link anchors. The application of link augmentation requires the existence of one or more linkbases that relate text anchors, i.e., sequences of words, to URLs. The architecture of link augmentation systems is similar to the architecture of open hypermedia systems as linking information is separated from the actual content of the hypertext pages. However, in the case of link augmentation systems, the selection of a linkbase is based on the needs of the user. Therefore, according to the navigational goals or interests of the user, a single page can be linked differently. Linkbases are usually predefined. An algorithm that automatically generates linkbases was presented by El-Beltagy et al. [28]. This algorithm extracts links from pages that the user has visited in the past and adds them to a Prolog knowledge base which acts as a linkbase. Link anchors are generated after segmenting the anchors of the original links and keeping words or phrases that directly relate to the content of the destination pages.

Map adaptation techniques have not been adequately investigated by AH systems. As we discussed in Section 2.6.3, maps and overview diagrams can assist the navigational tasks of hypertext users by contextualizing the browsed information and providing immediate access to individual pages in the hypertext system. An adaptive system would adapt map assistance based on the goals and interests of the user. Map adaptation could improve existing visualization techniques of hypertext overviews by (1) filtering information not relevant to the interests of the user, (2) highlighting and recommending parts of the overviews, and (3) visualizing semantic or structural relationships among the hypertext nodes according to the goals of the user. Tsandilas and schraefel [97] applied a type of map adaptation to visualize history lists. More specifically, they suggested history maps where both the time of access and the topics of the user's interests were used to lay out nodes on a 2-dimensional space. The nodes in these maps could be filtered by a set of sliders representing user interests. Two screenshots of the history maps are shown in Figure 3. The primary goal of the above approach was to increase the accessibility of history lists, which as mentioned in Section 2 are highly unused, by enabling users to easily adapt them to their needs.

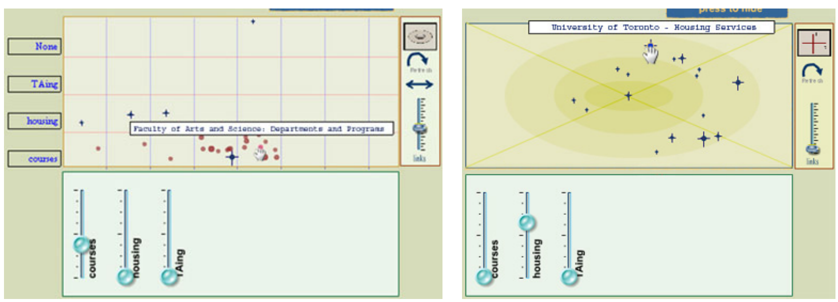


Figure 3. Adaptable maps of history lists

7 Usability Issues

- *What did you come in to look at?*
- *If you have any order to give me it's my duty to carry it out, he answered, after another silent pause, with a slow, measured lisp, raising his eyebrows and calmly twisting his head from one side to another, all this with exasperating composure.*

Notes from the Underground, Fyodor Dostoyevsky

Intelligent interfaces engage several problems and have received a lot of criticism [90]. Three major problems can be identified:

1. Adaptive systems depend on the construction of user models which are incomplete and usually erroneous.
2. They do not allow direct user control over the adaptation process.
3. They result in complex conceptual models which cannot be conceived by the users.

Due to the above problems as well as the lack of thorough evaluation studies, intelligence interfaces have failed to prove their usefulness. The very few intelligent interfaces that succeeded commercially have either performed very simple adaptations based on limited knowledge about the user or based adaptation on simple user actions rather than trying to infer complex user models [48]. Although research in the areas of UM and AI tries to address the first problem by applying new user modelling techniques and new intelligent mechanisms, it is commonly acknowledged that no user model can accurately describe a user. It is also hard to believe that future intelligent system will be able to precisely predict what a user wants as even human experts may fail to do so.

Intelligent interfaces often violate major usability principles established for direct-manipulation systems. The direct-manipulation paradigm was introduced in the early '80s as an alternative to dialog-based user interfaces and became prominent as graphical user interfaces emerged. Shneiderman [89] suggests three main principles describing direct manipulation:

1. Continuous representation of the objects and actions of interest.
2. Physical actions or presses of labelled buttons instead of complex syntax.
3. Rapid incremental and reversible operations whose effect on the object of interest is immediately visible.

These principles imply that the user should have direct control over the system, parts of the system should be transparent so that the user can understand something of its inner workings, and the reactions of the system should be predictable. Adaptive systems usually violate these principles as the system's output may unpredictably change based on assumptions or beliefs that are hidden from the user. In the following paragraphs, we examine the issues of transparency, predictability and user control in more detail.

7.1 Transparency and Predictability

The goal of building transparent systems is to help users build adequate mental models that correctly match the system's functionality. As Maes observes [90], transparency is an issue that applies not only to adaptive systems but to other systems as well. She argues that people do not always need to have complete knowledge of the internal parts of the system in order to be able to efficiently interact with the system. For instance, people can drive cars without having a complete model of how the engine or the brakes of the car work. Transparency, however, does not only refer to the visibility of the internal parts of a system but mainly to the visibility of the system's runtime behaviour and the context in which this behaviour is demonstrated. People can efficiently drive cars as long as they have good knowledge about the car's behaviour and reactions. A car provides direct feedback about the runtime state of the system so that the driver has adequate knowledge of the context in which the driving interaction takes place. For example, the car informs the driver about the speed of the car and the level of the fuel so that the driver can predict the car's reactions to his or her driving actions.

On the other hand, an adaptive system's behaviour may vary according to the details of the user model and its inference mechanism which are usually not transparent to the user. In some systems such as Letizia [61], the intelligent part of the user interface is separated from the predictable, non-intelligent part of the interface. In this way, users can consult the suggestions of the intelligent part without being continuously distracted by its unpredictable actions. Although his solution relaxes the problem by making the system's suggestions less intrusive, it does not improve the user's understanding of the adaptive behaviour of the system while it cannot generalize.

Cook and Kay [24] suggested that user models should be viewable. As user models can be complex and involve several parameters, the main challenge of this approach is the interpretation of the user model into a form that the user can easily understand. In their system, Cook and Kay provided visualizations of user models where the different parts of a user model were organized as interactive hierarchical structures. Different shapes were used to indicate the type of each node in the hierarchy, e.g., crosses represented user characteristics and diamonds represented user beliefs. The user could click on the nodes to unfold them and uncover its details. As Höök observes [48], depending on the application domain and the individual user's experience, it may be difficult to provide comprehensible views of user models. In this case, it may be appropriate to hide complex inference mechanisms from the user and show instead simplified views of the user model that provide a sense of predictability. Several learning systems have used "skillometers" to give an indication of a student model [54]. Skillometers enable the learners to see how the system models their progress. Other approaches [49, 56] have suggested the use of anthropomorphic agents that imitate human-human communication. These agents are gifted with facial expressions which provide a sort of transparency of what the agent believes about the user's goals. Shneiderman [90], on the other hand, argues against anthropomorphic agents as they give false expectations about their intelligence and their ability to communicate with the user.

The issues of transparency and predictability have gained only little attention by AH systems that support information seeking tasks. A representative example of such a system is PUSH [46], which based adaptation on a fixed set of task stereotypes which were viewable in a separate window. PUSH, however, was used in a specific domain which involved a constrained information space. Web navigation involves a large set of different domains, and tasks cannot be easily classified into fixed stereotypes. The techniques that we presented in Sections 4 and 5 employ more complex representations of user tasks and goals and it is not evident how they could be transformed into simple and viewable representations. For instance, several context-aware AH systems use large feature vectors to express the browsing context based on the content of documents that users have visited in the past. In this case, instead of overloading the user with the details of the underlying model, it would be most appropriate to provide a kind of simple and meaningful summarization of this information. It is also important to investigate visual representations of context which would give indications about the progress of the browsing process and somehow explain the adaptation result.

7.2 User Control

There is often made distinction between *adaptive* and *adaptable* user interfaces. In contrast to adaptive systems, adaptation in adaptable systems is determined mainly by the user and less by the system itself. Instead of requiring the designer of a system to make all the decisions at the design time, adaptable systems allow the actual user to make some decisions at use time [30] accommodating their own needs and preferences. The main advantage of adaptable systems against adaptive systems is that they give the users control over the process of adaptation and reduce the effects of incorrect system decisions. The cost of the increased controllability is the additional effort required from the user. The user may need to learn the adaptation component before being able to manipulate it. User control may have different forms and affect different levels of the system's adaptive behaviour. Figure 4 exhibits three different types of controllability in adaptable systems.

1. Users *customize* the interface by selecting the view which best satisfies their needs or select which functionality appears in the interface. The system does not provide any automatic assistance to support this task.
2. Users do not have direct control over the actual interface but they rather control the user model on which the system bases its adaptive behaviour.
3. Users control the level of the system's intrusiveness or the adaptation method.

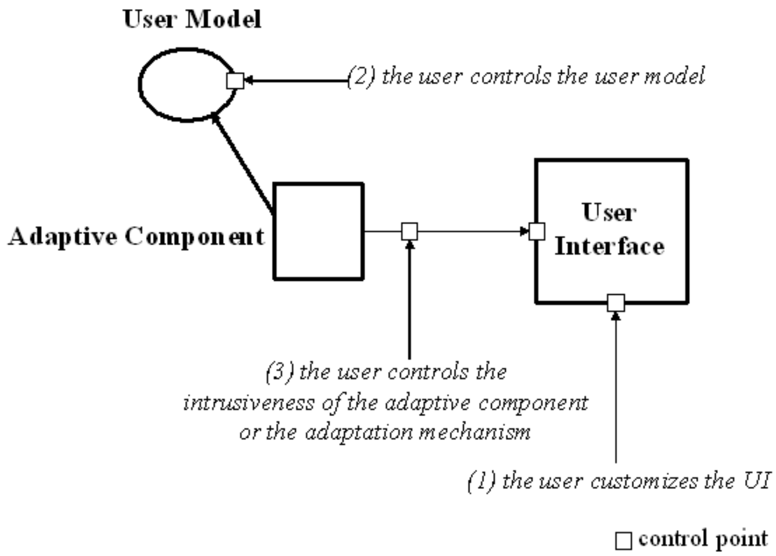


Figure 4. Forms of user control in adaptable systems

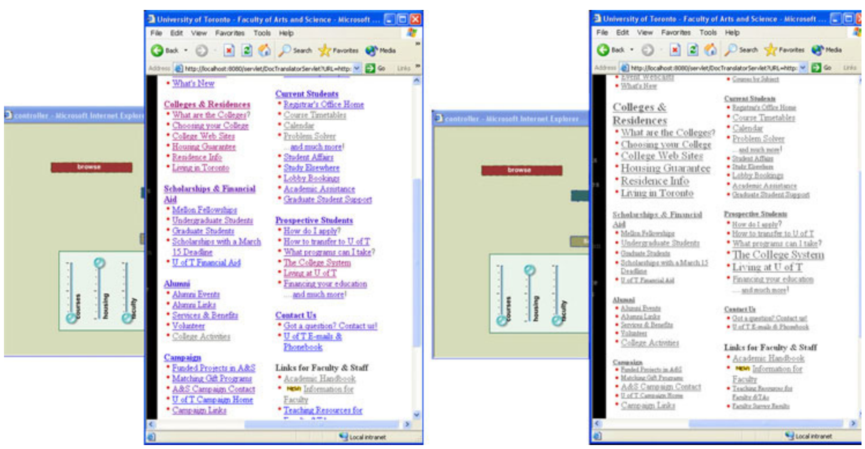
7.2.1 Customizing the User Interface

Comparing different versions of the commercial word processor Microsoft Word, McGrenere et al. [67] showed that customizable user interfaces may have advantages over interfaces which include adaptive features. However, their evaluation tested a specific adaptation mechanism and its results cannot generalize. Customizable interfaces often require the users to have advanced knowledge of the system, e.g., the user may need to set numerous parameters in a configuration file, while they cannot dynamically change as the needs of the users evolve. In the domain of hypertext, systems that belong to this category include Web sites with multiple views or hypermedia systems enhanced with interaction techniques that allow users to directly adjust the content of a page [86].

7.2.2 Controlling the User Model

The second type of adaptable interaction does not support direct control over the elements of the user interface. Customization is performed indirectly as the user controls the user model that the system preserves in order to decide on the final adaptation. This type of adaptability closely relates to the issue of transparency of user models; the user should be able to see aspects of the user model before applying changes to it. The hierarchical views that Cook and Kay [24] used to visualize user models enabled the application of several modifications on the underlying model. For instance, users could explicitly specify the value for a model component. Similarly, PUSH allowed users to explicitly specify and refine their tasks providing in this way a sort of controllability. Controllability of user models has been mainly investigated in the context of intelligent tutoring systems. Kay states that while early tutoring systems viewed users as *students*, the term *learner* is now favoured [54]. This implies that the role of the users is not passive, but they are responsible for their own learning, participating in the construction of their model and the selection of teaching strategies. Kay has introduced the notion of *scrutable* adaptive systems. Scrutable adaptive systems enable the user to investigate and review the way that the system has been adapted. Tutor [25] is a scrutable AH system which has been developed within this framework. At the beginning of each session, Tutor constructs a student model based on the answers of the student to a small set of profile questions. Based on this model, parts of the content may be excluded from a page. At the bottom of each page, there is a link to the explanation section. The explanation section explains how adaptation is performed and what content has been excluded from the page. Users can also revise their answers to the profile questions by clicking on an icon on the top of the page.

Scrutable adaptive systems do not necessary respect all the principles of direct manipulation. For instance, in the Tutor system, control operations are not incremental and are not immediately reflected to the adapted pages. Furthermore, although users can control the adaptation process, they have to switch to a different interaction mode to do so, which means that adaptation provided by the system and user control are not tightly coupled. Tsandilas and schraefel [97], on the other hand, proposed a scrutable system that eliminates the gap between adaptation and direct manipulation. According to this approach, user control is performed by means of interactive sliders whose manipulation has a direct effect on the result of the adaptation. More specifically, sliders represent particular topics of user interests and their continuous values declare how relevant to those topics the current user interests are. Consequently, a user model, whose internal representation is a feature vector, is expressed in terms of sliders, which have a clear metaphor as controllers. The position of the sliders determines the visualization of the links on a Web page, i.e., links are highlighted according to their relevance to the specified user interests. As discussed in Section 6, the link adaptation techniques that the above work applies exploit visual characteristic of the links such as colour and font sizes, which take continuous values. As a result of this, changes in the positions of the sliders are immediately translated into incremental changes in the visualization of the links. Two example snapshots that exhibit the above approach are shown in Figure 5.



(a) Adapting the colour (b) Adapting the font sizes

Figure 5. Direct manipulation of link adaptation

7.2.3 Controlling the Adaptation Method and the System's Intrusiveness

Adaptable systems that belong to the third category let the user control the adaptation method or the intrusiveness of the adaptation. For instance, commercial editors such as Microsoft Word provide options for specifying several parameters of the spelling checker. Research by Microsoft has tried to tackle the problem of balancing between automated assistance and intrusiveness and investigate how intelligence could be incorporated into direct manipulation interfaces. Horvitz refers to this type of interfaces as *mixed-initiative* user interfaces [49], in which users and intelligent agents collaborate to achieve the user's goals. The above work employs probabilistic models such Bayesian networks to capture the uncertainty about the goals of the user and infer the utility of the system's actions. Based on the various uncertainties and the cost of the different actions, the system may decide to take or not to take a particular action. Actions may involve system suggestions and tips, reminders or even questions aiming at resolving basic uncertainties.

Although the above approach seems to reduce the intrusiveness of adaptive systems by recognizing that user goals cannot always be predicted with certainty and that adaptive actions have a cost, it has some problems. Utilities, costs and connections between probabilistic parameters such as conditional probabilities in Bayesian networks, are often based on rough approximations or assumptions that do not hold. LookOut [49] addresses this problem by allowing users to explicitly specify utilities and threshold probabilities. Users, however, cannot get a clear picture of how these controls affect the behaviour of the intelligent system. The underlying probabilistic model is not transparent to the user, and as a result of this, the system's behaviour may appear as inconsistent and unpredictable.

An alternative to adjusting thresholds that determine when adaptation takes place is to adjust the actual effect of adaptation as perceived by the user. This suggests that the system's actions should always occur in a consistent manner, and what should be adapted and controlled is the intrusiveness of these actions. For instance, instead of deciding whether or not to show a suggestion, the system should decide on how large or detailed this suggestion would be; for example, the suggestion could appear as a small thumbnail at the corner of the window so that the user would not be distracted. In this way, the user would be able to anticipate the system's uncertainties and control them efficiently. The fisheye-like adaptation technique that we presented in Section 6.1.1 allows this form of controllable adaptation. The user can incrementally move from less to more adaptive views of the content in a page, and inversely, by simply manipulating a slider. No thresholds are defined since adaptation is based on adjusting the size of the various visual components in a page rather than displaying or hiding pieces of information.

8 Evaluations of AH systems

As Chin [21] states only about one third of articles on user models and user-adapted systems include any type of evaluation. As several evaluations are just pilot or informal studies, only a quarter of articles report significant results. Most evaluation approaches compare the adaptive system against its non-adaptive version. In Section 8.1, we present some well-known evaluations in the area of AH and information filtering systems. In Section 8.2, we discuss their limitations.

8.1 Evaluations

As mentioned in Section 4.3.2, IR measures such as precision, recall and accuracy are used to evaluate algorithms that learn user profiles. Most evaluations of adaptive systems are limited in testing the effectiveness of the underlying learning algorithm and their ability to correctly predict the needs of the users rather than testing the usability of the adaptive system. For instance, Joachims et al. [52] measured the performance of WebWatcher in terms of accuracy. Accuracy was defined as the percentage of the cases that users followed the advice given by the system. Results based on 5822 page tours with WebWatcher showed that in 43.9% of the cases where the system provided at least one and no more than three suggestions, the user followed the suggestion. This number was significantly smaller than 15.3% which was the calculated accuracy of a system which randomly highlighted links. However, as the authors of the above work state, this accuracy measurement is biased by the fact that the system's suggestions may influence the decisions of the users. In a different experiment, WebWatcher's performance was compared against the performance of eight human experts as well as the performance of randomly highlighting hyperlinks. The three ways of suggestion were applied after the completion of the page tours to predict three hyperlinks in a single page that each user was most likely to follow. The learning algorithm of WebWatcher achieved 42.9% accuracy compared to 22.4% accuracy of the random algorithm and 47.5% accuracy of the human experts.

Hirashima et al. [42] evaluated the predictive performance of their context-sensitive technique. They applied several different values of the decay parameter *r* to sort the titles of the candidate nodes. Nodes most likely to be relevant to the interests of the user were ordered higher in the list of candidate nodes. Applying a decay $r=0.6$ resulted in significantly better performance than setting *r* to *0* or *1* or keeping the order of the raw results. This result, however, may vary depending on the browsing behaviour of the individual users. If the user's interests change frequently, high values of *r* may result in better performance, while low values of *r* may perform better when the interests of the user change slowly or remain static. The above evaluation does not lead to any conclusions about the usability of reordering the recommendations based on the system's predictions. The users that participated in the evaluation did not anticipate any change in the reordering of the suggested nodes. Also, the evaluation did not measure how the performance of the users was affected by the various ordering methods.

A small number of approaches have attempted to evaluate the usability of the adaptive system that they proposed. Boyle and Encarnacion [10] conducted an experimental study to compare user performance in the case of three different systems: hypertext, stretchtext and MetaDoc, which is an adaptive system based on stretchtext. MetaDoc expands or collapses stretchtext “hot words” based on the user’s expertise, which is assumed to have two levels: novice and expert. The user’s expertise level and the system used were the independent variables of the evaluation of MetaDoc. Dependent variables were the time spent finding the correct answer, the number of correct answers, the number of nodes visited and the number of user commands. Subjects were tested on both reading comprehension and search/navigation questions. The results showed that MetaDoc users performed significantly lower times than stretchtext and hypertext users in reading comprehension tasks. The results also showed that stretchtext and MetaDoc were significantly better than hypertext in terms of the time spent for search and navigation questions as well as the number of correct answers.

Höök evaluated PUSH [47], which is also based on stretchtext. Höök argues that task completion time should not be the only criterion by which systems are evaluated. In some cases, it is the quality of the result rather than the time spent in completing a task that matters. Her evaluation tested nine subjects in several information seeking tasks. The results indicated that the adaptive system reduced the number of within-page actions. User satisfaction was also greater when using the adaptive system.

Finally, Jameson and Schwarzkopf [51] conducted an empirical study which illustrated the trade-offs between adaptation and controllability. They compared two different mechanisms of updating recommendations within an adaptive hotlist of a conference Web site: automatic versus controlled updating. In the controlled version, recommendations were updated after explicit requests from the user. 18 subjects participated in the above study. All the subjects tried both updating mechanisms. The main variable of interest was the users’ subjective evaluation of the two types of updating. The subjects’ responses did not lead to a general conclusion about which version was mostly preferred. Some subjects strongly preferred the controlled updating, whereas others strongly preferred the automatic updating. The study also showed that users may be willing to work with both versions of the system and switch between them as their tasks evolve over time.

8.2 Issues and Problems with Evaluations of AH Systems

As we mentioned earlier, most evaluations of adaptive systems compare the adaptive system against its non-adaptive version. According to Höök [47], the main problem of this approach is that the non-adaptive system may have not been optimally designed for the given task. If adaptivity is a natural part of the system, the evaluation may be biased against the non-adaptive system if the latter has not been carefully designed. In addition, an evaluation will be biased if the selection of the tasks that the subjects have to complete stresses the strengths of the adaptive system and the weaknesses of the non-adaptive system. An information seeking environment usually involves a large number of different tasks and different pieces of information. Selecting a small set of representative tasks is a difficult and tricky procedure.

Quality of answers, task completion time, number of actions and user satisfaction are the commonest measures used by evaluations of AH systems. Each of these measures cannot give clear information about the success of an adaptive system. For instance, an adaptive system may improve the average performance of users completing their tasks, but user satisfaction may decrease as the cost of incorrect guesses may be great or users may prefer to have absolute control over the system's actions. As discussed in Section 2, several researchers have proposed quantitative measures of hypertext disorientation, which should also be incorporated into the evaluations of AH systems. In addition to laboratory studies, field studies are also valuable. As McGrenere et al. [67] point out, users are more likely to express personalized behaviour when they do their own tasks in their normal work context rather than in a lab setting with prescribed tasks. There are no known field studies of AH systems.

Another important issue is that existing usability evaluations have not tried to control the accuracy of the intelligent part of the system. For instance, neither Höök [47] nor Boyle and Encarnacion [10] mention anything about the ability of the underlying learning mechanism to predict the goals of the user. Thus, the results of these evaluations cannot generalize to other systems or even to other tasks where the performance of the learning mechanism may vary. Future evaluation should distinguish between adaptation techniques and inference algorithms, so that it is possible to reach conclusions about how each of these factors affects the overall performance of an adaptive system.

Different groups of users may benefit differently from the adaptive features of a system. For example, novice users may find a system's recommendations useful while expert users may find them distracting. Some systems distinguish between the different user groups as their adaptation mechanism is based on this distinction. Other systems, however, do not make such a distinction. In this case, the evaluation should identify meaningful groups that respond differently to the adaptive features of the underlying system and investigate how the results vary among these different groups. As we mentioned in Section 8.1, Jameson and Schwarzkopf [51] observed large variations in the subjects' preferences over two different versions of an adaptive mechanism. This variation was not justified in terms of distinguishable characteristics of different user groups. On the other hand, McGrenere et al. [67] distinguished between "feature shy" and "feature keen" users in order to evaluate different versions of a word editor. Chin [21] suggests that if the separation into different user groups is not obvious, covariant variables that could affect the user performance should be measured by conducting pilot studies before the main experiment.

9 Conclusions and Future Directions

This paper reviewed research on adaptive hypermedia systems. It focused on systems that enhance and facilitate navigation and information finding on the Web. Research on hypertext navigation has identified several problems that hypertext users experience. Cognitive overhead and disorientation are the most frequently cited problems that affect navigation in a hypertext environment. Besides, information overload, which is a well-known problem in today's Web, increases the effects of disorientation and cognitive overhead and decreases the effectiveness of information finding tasks. Navigational and information finding tools such maps, history lists, glosses and search engines relax the above problems, but they engage several limitations. AH systems address some of the limitations of these tools by providing automated assistance based on the information needs and goals of an individual user. AH systems represent user needs and goals by user models which are learnt based on implicit or explicit user feedback. The paper reviewed several approaches that infer the user model based on the content of the pages that the user visits. Special attention was given to systems which aim at capturing the underlying navigation context and specifying how this context evolves over time.

User information captured by an AH system determines how hypertext pages are adapted. Minimizing the information overload, facilitating navigation by providing hints and suggestions, and recommending related sources of information form the main types of assistance that AH systems provide. Adapting the content or the linking information within a page may be useful but it entails several dangers: important landmarks in the page may be removed; the surrounding context of the information may be lost; users may be deprived of the chance to discover new topics of interest and refine their goals. We claimed that merging focus+context visualization techniques such as fisheye views with adaptation techniques could relax these problems. Then, adaptation could be seen as a process of synchronizing the focus of the projected information with the focus of the user's interests. As the interests of the user evolve, the focus of the adaptation moves while context is preserved. We presented some initial work [96] towards this direction. We plan to extend this work to capture more general page layouts and further evaluate it with respect to other existing adaptation techniques. An interesting question that needs to be answered is how focus and context should be balanced in focus+context adaptation. We are interested in investigating how adaptation factors such as the accuracy of the user model or the learning algorithm should affect this balance. It is also part of our goals to examine how focus+context adaptation could be useful in domains other than hypertext.

Due to serious usability problems that adaptivity engages, there have been a few only successful adaptive systems. Most of the problems originate from the fact that user models are imperfect and learning algorithms have limitations, mainly because there is usually poor evidence about the goals of a user. Some approaches suggest that the user should be able to access the user model and eliminate wrong beliefs or uncertainties. In this review, we stressed the need of bridging the gap between adaptation and direct manipulation. This implies that user models should become transparent and controllable. It also means that manipulations of a user model should be reflected as manipulations of the visual components of the user interface. We suggest that the interaction between a user and an adaptive interface should be performed in two different complementary ways: (1) through interaction with the elements of the user model, and (2) through direct interaction with the main elements of the user interface. We believe that these two types of interaction should be tightly coupled allowing the cooperative construction of rich interaction contexts. Changes in the user model should be mapped to changes in the main elements of the user interface. Inversely, as the user interacts with the user interface, changes in the context of the interaction should become immediately visible through a comprehensive visual representation of the user model. Merging the above two types of interaction under the same interaction model is a challenging research problem.

Finally, future research on AH systems and adaptive systems, in general, should pay more attention on conducting evaluations whose results could be generalized in other systems and other situations. Numerous AH systems have been developed, focusing on a variety of problems and employing several different techniques. Existing research has not attempted to provide a common framework of comparison among different systems and different aspects of adaptation.

References

1. Abrams, D., R. Baecker, and M. Chignell. *Information Archiving with Bookmarks: Personal Web Space Construction and Organization*. In *CHI'98 Conference on Human Factors in Computing Systems*. 1998. Los Angeles CA, United States, ACM Press.
2. Ahuja, J.S. and J. Webster, *Perceived disorientation: an examination of a new measure to assess web design effectiveness.* Interacting with Computers, 2001. **14**(1): p. 15-29.
3. Bailey, C., S.R. El-Beltagy, and W. Hall. *Link Augmentation: A Context-Based Approach to Support Adaptive Hypermedia.* In *3rd Workshop on Adaptive Hypertext and Hypermedia*. 2001. Aarhus, Denmark.

4. Balabanovic, M. and Y. Shoham. *Learning Information Retrieval Agents: Experiments with Automated Web Browsing*. In *AAAI Spring Symposium on Information Gathering from Heterogenous, Distributed Resources*. 1995. Stanford, CA.
5. Bauer, T. and D.B. Leake. *WordSieve: A Method for Real-Time Context Extraction*. In *3rd International and Interdisciplinary Conference, Context*. 2001. Dundee, Scotland, Springer Verlag.
6. Bederson, B.B. and J.D. Hollan. *Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics*. In *User Interface and Software Technology, UIST'94*. 1994. Marina Del Ray, California, USA, ACM Press.
7. Bederson, B.B., et al., *A Zooming Web Browser*, In *Human Factors and Web Development*, N.J.L. Erlbaum, Editor. 1998. p. 255-266.
8. Bergman, M.K., *The deep Web: Surfacing Hidden Value.* The Journal of Electronic Publishing from the University of Michigan, 2001.
9. Berners-Lee, T., J. Hendler, and O. Lassila, *The Semantic Web.* Scientifiic American, 2001. p. 28-37.
10. Boyle, C. and A.O. Encarnacion, *MetaDoc: An Adaptive Hypertext Reading System.* User Modeling and User-Adapted Interaction, 1994. **4**(1): p. 1-19.
11. Broder, A., et al. *Graph structure in the web*. In *9th International World Wide Web Conference*. 2000. Amsterdam, Holland.
12. Broekstra, J., et al. *Enabling knowledge representation on the Web by extending RDF schema*. In *10th International World Wide Web Conference*. 2001. Hong Kong, China, ACM Press.
13. Brusilovsky, P., *Methods and Techniques of Adaptive Hypermedia.* User Modeling and User-Adapted Interaction, 1996. **6**(2-3): p. 87-129.
14. Brusilovsky, P., *Adaptive Hypermedia.* Modeling and User Adapted Interaction, 2001. Ten Year Anniversary Issue **11**(1/2): p. 87-110.
15. Budzik, J., H. K., and L. Birnbaum, *Information access in context.* Knowledge based systems, 2001. **14**(1-2): p. 37-53.
16. Bush, V., *As We May Think.* The Atlantic Monthly, 1945. **176**(1): p. 101-108.
17. Canter, D., R. Rivers, and G. Storrs, *Characterizing User Navigation Through Complex Data Structures.* Behaviour and Information Technology, 1985. **4**(3): p. 93-102.
18. Carr, L., S. Bechhofer, C. Goble, and W. Hall. *Conceptual Linking: Ontology-based Open Hypermedia*. In *the 10th International World Wide Web Conference*. 2001. Hong Kong, China.
19. Catledge, L.D. and J.E. Pitkow. *Characterizing Browsing Strategies in the World-Wide Web*. In *3rd International World Wide Web Conference*. 1995. Darmstadt, Germany.
20. Chen, C., M. Czerwinski, and R. Macredie, *Individual Differences in Virtual Environments: Introduction and Overview.* Journal of the American Society of Information Science, 2000. **51**(6): p. 499-507.
21. Chin, D., *Empirical Evaluations of User Models and User-Adapted Systems.* User Modelig and User-Adapted Interaction, 2001. Ten Year Anniversary Issue.
22. Cockburn, A. and S. Greenberg. *Issues of Page Representation and Organisation in Web Navigation Tools*. In *OZCHI'99 Australian Conference on Human Computer Interaction*. 1999. Wagga Wagga, Australia.
23. Conklin, J., *Hypertext: an introduction and survey.* Computer, IEEE Computer Society Press, 1987. **20**(9): p. 17-41.
24. Cook, R. and J. Kay. *The justified user model: a viewable, explained user model*. In *4th International Conference on User Modeling*. 1994. Hyannis, MA.
25. Czarkowski, M. and J. Kay. *How to give the user a sense of control over the personalization of AH?* In *Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2003, User Modeling Session*. 2003.
26. Davis, H., W. Hall, I. Heath, G. Hill, and R. Wilkins. *MICROCOSM: An open hypermedia environment for information integration*. In *ECHT*. 1992.
27. Edmunds, A. and A. Morris, *The problem of information overload in business organisations: a review of the literature.* International Journal of Information Management, 2000. **20**(1): p. 17-28.

28. El-Beltagy, S.R., W. Hall, D.D. Roure, and L. Carr. *Linking in Context*. In *12th ACM conference on Hypertext and Hypermedia*. 2001. Erhus, Denmark, ACM Press.
29. Farris, J.S., K.S. Jones, and P.D. Elgin, *Users' schemata of hypermedia: What is so spatial about a website?* Interacting with Computers, 2002. **14**: p. 487-502.
30. Fischer, G., *User Modeling in Human-Computer Interaction.* User Modeling and User-Adapted Interaction, 2001. **11**(1&2): p. 65-86.
31. Foss, C., *Tools for reading and browsing hypertext. Information Processing and Management.* Information Processing and Management, 1989. **25**(4): p. 407-418.
32. Fung, R. and B.D. Favero, *Applying Bayesian networks to information retrieval.* Communications of the ACM, 1995. **38**(3): p. 42-ff.
33. Furnas, G.W. *Generalized Fisheye Views*. In *CHI '86*. 1986. Boston, Massachussetts, USA, ACM Press.
34. Furnas, G.W. and J. Zacks. *Multitrees: enriching and reusing hierarchical structure*. In *SIGCHI '94 Conference on Human Factors in Computing Systems: "celebrating interdependence"*. 1994. Boston, Massachusetts, USA, ACM Press.
35. Giunchiglia, F. and P. Bouquet. *A Context-Based Framework for Mental Representation*. In *20th Annual Meeting of the Cognitive Science Society, CogSci'98.* 1998. Wisconsin, Madison, US.
36. Goecks, J. and J. Shavlik. *Learning users' interests by unobtrusively observing their normal behavior*. In *5th International Conference on Intelligent User Interfaces*. 2000. New Orleans, Louisiana, US, ACM Press.
37. Green, S.J. *Building hypertext links in newspaper articles using semantic similarity*. In *Third Workshop on Application of Natural Language to Information Systems (NLDB '97).* 1997. Vancouver, Canada.
38. Greenberg, S., C. Gutwin, and A. Cockburn. *Using Distortion-Oriented Displays to Support Workspace Awareness*. In *People and Computers XI (Proceedings of HCI '96).* 1996, Springer-Verlag.
39. Halasz, F. and M. Schwartz, *The Dexter hypertext reference model.* Communications of the ACM, 1994. **37**(2): p. 30-39.
40. Head, M., N. Archer, and Y. Yuan, *World Wide Web Navigation Aid.* International Journal of Human-Computer Studies, 2000. **53**(2): p. 301-330.
41. Hightower, R.R., L.T. Ring, J. Helfman, B.B. Bederson, and J.D. Hollan. *Graphical Multiscale Web Histories: A Study of PadPrints*. In *9th ACM Hypertext and Hypermedia Conference*. 1998. Pittsburgh, PA, USA, ACM Press.
42. Hirashima, T., N. Matsuda, T. Nomoto, and J. Toyoda. *Context-sensitive filtering for browsing in hypertext*. In *3rd International Conference on Intelligent User Interfaces*. 1998. San Francisco, California, US, ACM Press.
43. Hirst, G. *Context as a spurious concept.* In *AAAI Fall Symposium on Context in Knowledge Representation and Natural Language*. 1997. Cambridge, MA.
44. Hohl, H., H.-D. Bφcker, and R. Gunzenhδuser, *HYPADAPTER: An Adaptive Hypertext System for Exploratory Learning and Programming.* User Modeling and User-Adapted Interaction, 1996. **6**(2-3): p. 131-155.
45. Holmquist, L.E. *Focus+Context Visualization with Flip Zooming and the Zoom Browser*. In *Proceedings of ACM CHI '97 Conference on Human Factors in Computing Systems - Extended Abstracts*. 1997, ACM Press.
46. Hook, K., et al., *A Glass Box Approach to Adaptive Hypermedia.* User Modeling and User-Adapted Interaction, 1996. **6**(2-3): p. 157-184.
47. Hook, K. *Evaluating the Utility and Usability of an Adaptive Hypermedia System*. In *International Conference on Intelligent User Interfaces, IUI 97.* 1997. Orlando, Florida, USA, ACM Press.
48. Hook, K., *Steps to take before Intelligent User Interfaces become real.* Interacting with Computers, 2000. **12**(4): p. 409-426.

49. Horvitz, E. *Principles of Mixed-Initiative User Interfaces*. In *ACM SIGCHI Conference on Human Factors in Computing Systems*. 1999. Pittsburgh, PA.
50. Hothi, J.K., W. Hall, and T. Sly. *A Study Comparing the Use of Shaded Text and Adaptive Navigational Support in Adaptive Hypermedia*. In *International Conference of Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2000*. 2000. Trento, Italy.
51. Jameson, A. and E. Schwarzkopf. *Pros and Cons of Controllability: An Empirical Study*. In *Adaptive Hypermedia and Adaptive Web-Based Systems. Second International Conference, AH2002*. 2002. Malaga, Spain, Springer.
52. Joachims, T., D. Freitag, and T.M. Mitchell. *WebWatcher: A tour guide for the World Wide Web*. In *IJCAI*. 1997.
53. Kaplan, C., J. Fenwick, and J. Chen, *Adaptive hypertext navigation based on user goals and context.* User Modeling and User-Adapted Interaction, 1993. **3**: p. 193-220.
54. Kay, J., *Learner Control.* User Modelig and User-Adapted Interaction, 2001. **11**(1/2): p. 111-127.
55. Keep, C., T. McLaughlin, and R. Parmar, *The Electronic Labyrinth*, http://www.iath.virginia.edu/elab/elab.html.
56. Koda, T. and P. Maes. *Agents with Faces: The Effects of Personification of Agents*. In *HCI'96*. 1996. London, UK.
57. Kushmerick, N., J. McKee, and F. Toolan. *Towards Zero-Input Personalization: Referrer-Based Page Prediction*. In *International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, AH 2000*. 2000. Trento, Italy.
58. Lamping, J., R. Rao, and P. Pirolli. *A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies*. In *ACM CHI'95 Conference on Human Factors in Computing Systems*. 1995. Denver, Colorado, USA, ACM Press.
59. Lewis, D.D. *Naive Bayes at forty: The independence assumption in information retrieval.* In *ECML-98, 10th European Conference on Machine Learning*. 1998, Springer Verlag.
60. Li, W.-S., Q. Vu, D. Agrawal, Y. Hara, and H. Takano. *PowerBookmarks: A System for Personalizable Web Information Organization, Sharing, and Management*. In *8th International World Wide Web Conference*. 1999. Toronto, Canada.
61. Lieberman, H. *Letizia: An Agent That Assists Web Browsing*. In *15th International Joint Conference on Artificial Intelligence, IJCAI-95*. 1995. Montreal, Quebec, Canada, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.
62. Lynch, K., *The Image of the City*. 1960, Cambridge, Massachusetts: The MIT Press.
63. Marchionini, G., *Information Seeking in Electronic Environments*. Cambridge Series on Human-Computer Interaction. 1995: Cambridge University Press.
64. McCarthy, J., *Generality in Artificial Intelligence.* Communications of ACM, 1987. **30**(12): p. 1030-1035.
65. McDonald, S. and R.J. Stevenson, *Disorientation in hypertext: The effects of three text structures on navigation performance.* Applied Ergonomics, 1996. **27**(1): p. 61-68.
66. McGowan, J., N. Kushmerick, and B. Smyth. *Who do you want to be today? Web Personae for personalised information access.* In *Adaptive Hypermedia and Adaptive Web-Based Systems. Second International Conference, AH 2002*. 2002. Malaga, Spain, Springer Verlag.
67. McGrenere, J., R.M. Baecker, and K.S. Booth. *An evaluation of a multiple interface design solution for bloated software*. In *ACM CHI 2002 Conference on Human Factors in Computing Systems*. 2002. Minneapolis, Minnesota, USA, ACM CHI Letters.
68. McKenzie, B. and A. Cockburn. *An empirical analysis of web page revisitation*. In *the 34th Hawaiian International Conference on System Sciences, HICSS34*. 2001. Maui, Hawaiii, IEEE Computer Society Press.
69. Mladenic, D., *Personal WebWatcher: Implementation and Design.* Technical Report IJS-DP-7472. 1996, Department of Intelligent Systems, J.Stefan Institute.: Slovenia.
70. Mladenic, D., *Text-learning and related intelligent agents.* IEEE EXPERT, 1999. (Special Issue on Applications of Intelligent Information Retrieval).

71. Modjeska, D., *Hierarchical Data Visualization in Desktop Virtual Reality*. Ph.D. thesis. *University of Toronto*. 2000.
72. Nelson, T., *PROJECT XANADU*, http://www.xanadu.net/.
73. Nelson, T., *ZigZag Software, Design for a new Computer Universe*, http://xanadu.com/zigzag/.
74. Nielsen, J., *Multimedia and Hypertext: the Internet and Beyond*. 1995: Academic Press Professional.
75. Noik, E.G. *Exploring large hyperdocuments: fisheye views of nested networks*. In *Proceedings of the 5th ACM conference on Hypertext*. 1993. Seattle, Washington, United States, ACM Press.
76. Norman, D.A., *Some Observations on Mental Models*. Mental Models, ed. D. Gentner and A.L. Stevens. 1986: Hillsdale, New Jersey: Lawrence Erlbaum Associates.
77. Otter, M. and H. Johnson, *Lost in Hyperspace: metrics and mental models.* Interacting with computers, 2000. **13**(1): p. 1-40.
78. Passini, R., *Wayfinding in Architecture*. 1984, New York: Van Nostrand Reinhold.
79. Pazzani, M.J., J. Muramatsu, and D. Billsus. *Syskill & Webert: Identifying Interesting Web Sites*. In *13th National Conference on Artificial Intelligence*. 1996. Portland, OR, US.
80. Pirolli, P. and S. Card, *Information Foraging.* Psycological Review, 1999. **106**(4): p. 643-675.
81. Rhodes, B.J. *Margin Notes: Building a Contextually Aware Associative Memory*. In *4th International Conference on Intelligent User Interfaces*. 2000. New Orleans, LA USA, ACM Press.
82. Robertson, G.G., J.D. Mackinlay, and S.K. Card. *Cone Trees: animated 3D visualizations of hierarchical information*. In *ACM CHI '91 Conference on Human Factors in Computing Systems*. 1991. New Orleans, LA, USA, ACM Press.
83. Robertson, G.G., et al. *Data Mountain: Using Spatial Memory for Document Management*. In *ACM UIST '98 Symposium on User Interface Software and Technology*. 1998. San Francisco, CA, ACM Press.
84. Rollason, C., *Borges' Library of Babel and the Internet*, In *The Modern Word*. 1999.
85. Salton, G., *Developments in automatic text retrieval.* Science, 1991. **253**: p. 974-979.
86. schraefel, m.c. *ConTexts: Adaptable Hypermedia*. In *Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000*. 2000. Trento, Italy, Springer-Verlag.
87. schraefel, m.c., Y. Zhu, D. Modjeska, D. Wigdor, and S. Zhao. *Hunter gatherer: interaction support for the creation and management of within-web-page collections*. In *the 11th International World Wide Web Conference, WWW2002*. 2002. Honolulu, Hawaii, USA.
88. Sebastiani, F., *Machine learning in automated text categorization.* ACM Computing Surveys, 2002. **34**(1): p. 1-47.
89. Shneiderman, B., *Direct manipulation: A step beyond programming.* IEEE Computer, 1983. **16**(8): p. 57-69.
90. Shneiderman, B. and P. Maes, *Direct manipulation vs. interface agents.* Interactions, ACM, 1997. **4**(6): p. 42-61.
91. Skopik, A. and C. Gutwin. *Finding Things In Fisheyes: Memorability in Distorted Spaces*. In *Conference on Graphics Interface GI'03*. 2003. Halifax, Canada.
92. Smith, P.A., *Towards a practical measure of hypertext usabilty.* Interacting with Computers, 1996. **8**(4): p. 365-381.
93. Stanyer, D. and R. Procter. *Improving Web usability with the link lens*. In *8th International WWW Conference*. 1999. Toronto, Canada.
94. StatMarket, *Branding on the Web Takes Hold as Web "Surfing" Steadily Declines*. 2002,http://www.statmarket.com.
95. Tauscher, L. and S. Greenberg, *How People Revisit Web pages: Empirical Findings and Implications for the Design of History Systems.* International Journal of Human-Computer Studies, 1997. **47**: p. 97-137.
96. Tsandilas, T. and m.c. schraefel. *Adaptive Presentation Supporting Focus and Context*. In *Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2003*. 2003. Nottingham, UK.
97. Tsandilas, T. and m.c. schraefel. *User-Controlled Link Adaptation*. In *the 14th ACM Conference on Hypertext and Hypermedia, HT'03*. 2003. Nottingham, UK, ACM Press.

98. Vinson, N.G. *Design Guidelines for Landmarks to Support Navigation in Virtual Environments*. In *CHI '99*. 1999. Pittsburgh, PA, ACM Press.
99. W3C, *XML Pointer, XML Base and XML Linking.*
100. Wong, S.K.M. and C.J. Butz, *A Bayesian Approach to User Profiling in Information Retrieval.* Technology Letters, 2000. **4**(1): p. 50-56.
101. Woods, D.D., *Visual Momentum: A Concept to Improve the Cognitive Coupling of Person and Computer.* International Journal of Man-Machine Studies, 1984. **21**(3): p. 229-244.
102. Yang, Y., S. Slattery, and R. Ghani, *A Study of Approaches to Hypertext Categorization.* Journal of Intelligent Information Systems, 2002. **18**(2-3): p. 219-241.
103. Zellweger, P.T., B.-W. Chang, and J.D. Mackinlay. *Fluid links for informed and incremental link transitions*. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia : links, objects, time and space-structure in hypermedia systems*. 1998. Pittsburgh, Pennsylvania, United States, ACM Press.
104. Zhu, E., *Hypermedia Interface Design: The Effects of Number of Links and Granularity of Nodes.* Journal of Educational Multimedia and Hypermedia, 1999. **8**(3).