Chapter 3

Literature Review

3.1 Introduction

This chapter investigates methods of actively supporting individual users in their quest for information, in hypertext and information retrieval systems. In particular, we will look at methods which strive to reduce the complexity of the information space to make finding information in it more tractable to the user.

We begin by discussing general hypertext[1] and information retrieval systems. We will compare hypertext and information retrieval systems on four main points, namely, the structure, representation, organisation of information, and the interaction the user has with each system. We then present adaptive hypertext systems, and describe the support structures required to make them possible.

3.1.1 General-purpose hypertext systems

A hypertext is a collection of nodes and hyperlinks (links). Links connect nodes together in the form of a network (figure 3.1). A node has zero or more in-links (a node with no in-links is called a *root node*), and zero or more out-links (a node that has no out-links is a *leaf node*). All nodes other than root nodes have parents, identified by the node's in-links, and all nodes other than leaf nodes have children, identified by the node's out-links.

---

[1] We use hypertext and hypermedia interchangeably throughout this thesis.

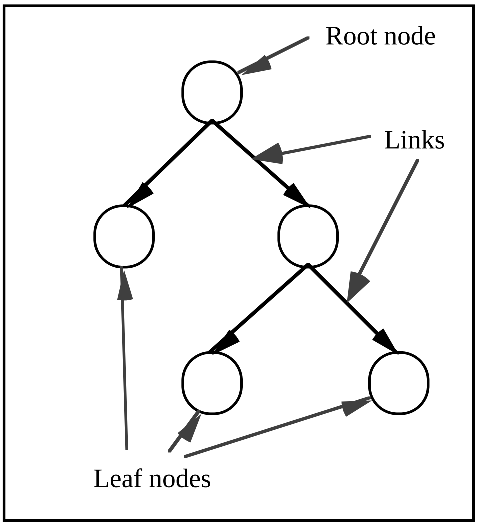


Figure 3.1: A simple hypertext

Nodes are linked usually because the child node is relevant to the parent node in some way. A *link anchor* in the parent usually provides a visual indicator that it is the source of a link. The user can access the node at the destination of the link by clicking on the link anchor, or by performing some action which results in the node at the destination of the link being displayed to the user. In general-purpose hypertext systems there are no firm rules for linking nodes, and node authors can create links between irrelevant documents. However, in order for the organisation of information in the hypertext to be meaningful to the user, links should lead to related information. A strict implementation of this guideline is likely to result in a hierarchical tree-like hypertext structure, where each child provides more specific information about some detail of the parent. In the loosest sense, nodes can be linked so arbitrarily that children will have little or no relevance to their parents.

A user interacts with a hypertext network by starting from a known (or default) node and subsequently traversing links. Nodes may also be directly accessed via their unique handles (the Uniform Resource Locator, in the case of the World Wide Web [4]). The latter form of interaction requires users to know *a priori* a node's handle. In the absence of this information, the user must rely on the hypertext's implicit semantic organisation and hope that the hypertext's author has made it reasonably obvious how information in the hypertext has been organised. A hypertextual classification system, in which sub-categories provide finer and finer levels of detail, best suits this type of interaction if the user knows, or can determine, to which sub-category the sought information belongs. It is possible for a user to develop an accurate mental model of a small hypertext or portions of a large hypertext, if the hypertext is visited frequently and changes infrequently. In this case, the user may learn which nodes contain what information, and may select nodes to act as landmarks for accessing nearby nodes, and then access such landmark nodes directly through their handles. In terms of our town-dweller metaphor, after initially having to exhaustively walk the streets, the recently arrived inhabitant navigates around town by learning points of reference, and by discovering that having sight of a particular landmark means that the sought after establishment is just around the corner. In a larger town, the landmarks remembered may cover greater and greater areas, they become further removed from the detail required (a higher level of abstraction) and may be used to provide a point of reference to a number of different types of services.

In large hypertexts, random or poor associations of nodes via links may make it difficult for human users to build a reliable mental model of the organisation of information contained in the hyperspace [89]. On the other hand, utilising standard guidelines to control how and when links are created between nodes may make it significantly easier for users to maintain a reliable mental model to support efficient navigation strategies ([88], [37]). Usually, flexible hypertexts are organisations of information which fall somewhere in between the two extremes.

Links can be unidirectional or bi-directional. If a link is unidirectional, it is possible to follow a link from its source to its destination, but it is impossible for the link to be followed from its destination to its source. Consider the World Wide Web: when a node (Web page) is accessed, it is possible for the visitor to follow any of the Web page's links. However, it is impossible for the visitor to access any of the Web pages which contain the link source to that page via the link (other than the page containing the link that they followed).

Although few hypertext systems support it, links may also be multi-headed (for example, [45]). The process of following a multi-headed link may involve the user selecting one of a number of different destinations. In terms of our town-dweller, the difference between unidirectional and bi-directional links would be equivalent to the difference between one-way and two-way streets (which even pedestrians must respect!). Multi-headed links, however, do not yet have a direct parallel in our metaphor. If street corners are nodes and streets are links, then a multi-headed link would be similar to a street being capable of transporting the user to one of a number of different street corners.

A number of formal models for hypertext systems exist. One of these, the Dexter Hypertext Reference Model (DHRM) [45], has influenced implementations of hypertext systems, as well as the HyperContext framework. DHRM separates a hypertext system into three layers: the Run-Time Layer, the Storage Layer, and the Within-Component Layer. Chapter 4.2 of this thesis explains the relationship between DHRM and HyperContext, but here we give a brief description of DHRM.

DHRM focuses mainly on the storage layer which models the basic nodes/link network structure of a hypertext. The within-component layer is concerned with the structure and contents of nodes. To allow future document types to be supported, and to allow linking across documents of different type, Dexter does not describe a generic model of the within-component layer. The run-time layer is responsible for the presentation of the hypertext to the user and user interactions with a hypertext. Interaction between the storage and within-component layers is achieved through a process called *anchoring*. Anchoring defines how a component or a region within a component can be linked to another component or region.

The major implications of DHRM are that hyperlinks (links) are stored separately from the actual documents representing nodes in hyperspace and that document contents can be modified just before presentation to the user. DHRM also makes use of "resolver" and "accessor" functions, to identify a "component" (node) by its "specification" (description) and to access the node respectively. Links in DHRM can be multi-headed (multiway), because the resolver function may return a number of components which all match the required specification.

DHRM has been implemented in the Amsterdam Hypermedia Model, which adds time and context to the model [46], and DeVise which extends the model to support cooperative work [43].

As a hypertext is a collection of nodes and links, together forming a network, it is possible to present users with a graphical overview of the network neighbourhood, and to give the users the ability to change the hierarchical detail at which they view the network. A common criticism of large hypertexts is that users experience the "Lost in Hyperspace" syndrome in which they have lost perspective of where they are in relation to the information around them, especially if the document's author has not provided navigational links to significant or familiar landmarks ([88], [37]). A map which shows the user's current position in the hypertext would be a significant aid. However, in hypertexts with a significant number of nodes and links, it is not always obvious how to give the user just the right level of detail to be able to re-orient herself without overwhelming her (for example, [67]).

The Open Hypermedia community promote the use of external link databases (for example, [21]) to store linking information separately from the resources (e.g., documents) that they link to and from. The main motivations for open hypermedia are to support hypertextual linking between non-editable objects and for link re-use and maintenance. A side-effect of storing links separately from documents and binding links to documents at presentation time is that links and link destinations can be modified immediately prior to presentation of the document to the user.

Microcosm [27] was one of the first hypertext systems to be constructed around the concept of externally stored links. In Microcosm, links may be described as end-points which are relevant to a source document in certain situations. For example, a *specific* link may define a specific source document and source anchor location. Alternatively, a *local* link may define a specific source document, but the link may be traversed from any point within that document. Finally, a *generic* link may define when it should be available in any source document, and it will automatically bind to a source document whenever the conditions are satisfied. In these cases, the link destination is static. Microcosm also supports dynamically computed link destinations, using text retrieval techniques.

3.1.2 General-purpose Information Retrieval systems

An Information Retrieval (IR) system takes a user query and returns a list or set of documents which relate to the query. A full-text search through all documents in a massive document space would normally approach computational intractability, so documents are usually represented by a structure that supports a more tractable search strategy. This structure is usually an *inverted index* which contains words (*index terms*) extracted from all documents, followed by the identities of the documents which contain each index term. Finding relevant documents can involve extracting the *postings list* (the set of documents) from the inverted index for each term in the query, and performing set operations on them to determine the documents that lie at the intersection of the sets ([90], Chapter 5).

Information Retrieval is used by users who can describe what they are looking for, but do not know where it exists. Our town-dweller would require such a service if she does not know the address of an establishment, but can describe, for example, the type of business it conducts. Both the town-dweller and the information retrieval user require that the person or system being queried is familiar with the domain. A town-dweller may describe in precise terms the establishment required, but if the person asked is also a stranger to the town, then the response will be unsatisfactory. The level of representation that an information retrieval system has of the documents in its document space is similarly significant.

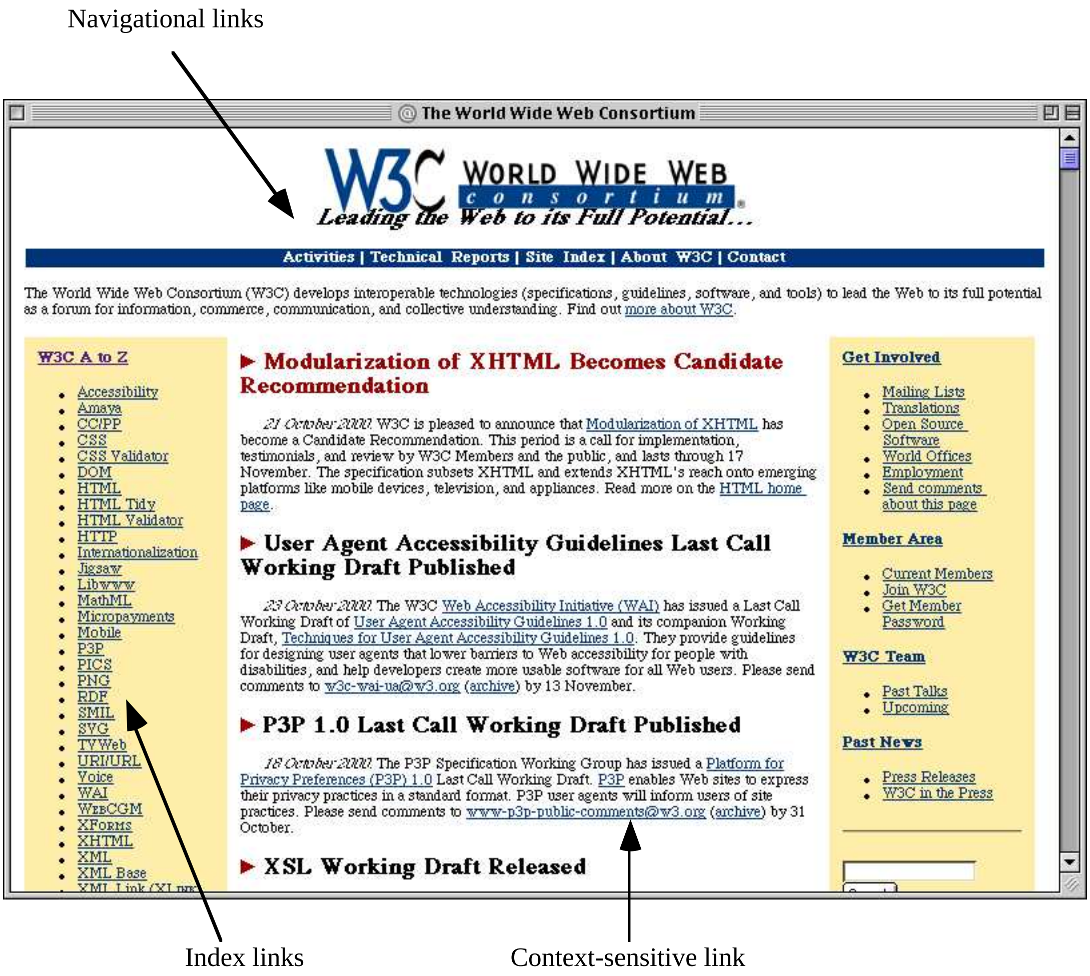
There are numerous matching algorithms to compare a query with documents which variously suit domain-independent document collections to collections which are domain-specific (see [90], [66], [78]). Our present concern is with general-purpose information retrieval systems which operate with domain-independent document collections. A characteristic of such environments is that its users are various and the information contained in the document collection is heterogeneous. This characteristic is shared with general-purpose hypertext systems. Users will have disparate information requirements and users with different levels of expertise, both with the system, as well as with the information contained in the document collection, will use the system. The same document in the collection may be relevant to different users with different information requirements, and users with similar queries may require information contained in different documents. For example, a report on a football match between Liverpool F.C. and Manchester United F.C. may be equally relevant to users looking for information on Liverpool F.C., as well as those who are looking for information on Manchester United F.C. Conversely, whereas one user may consider the report as sufficient evidence of the great rivalry between these two football clubs, another user may not.

During the transformation process of deriving a representation from a document, there is usually a loss of information. This can result in a number of different documents sporting the same representation, which can also lead to users with different queries discovering the same document is relevant. Equally, users with similar needs can have a difference of opinion as to the relevance of the same document to their individual queries. To compensate, information retrieval system users sometimes are able to give relevance feedback on documents which have been retrieved. This enables the information retrieval system to search for other documents based on the feedback the user has given, by reformulating the query given the representations of documents the user considers relevant and non-relevant [74].

Consider two town-dwellers looking for a bakery. They (separately) approach the same person (the local know-it-all!) to ask the whereabouts of the local bakery. They are directed to the same bakery, which is just a short walk away. One of the town-dwellers, after sampling the bakery's speciality, is perfectly satisfied. The second, however, finds the bread a little on the dry side and seeks out the know-it-all to ask for another bakery. The know-it-all will attempt to extrapolate information from the fact that the bakery was not suitable in order to identify another bakery which is likely to be more acceptable. If the town-dweller is patient enough, then after a number of expeditions between the know-it-all and the various bakeries, she may either find a suitable bakery, or else she will determine that the ideal bakery does not exist in this town. Incidentally, the town-dweller may also realise that she can be a bit more specific about her requirements, so apart from allowing the know-it-all to select a different bakery based on the fact that she did not quite like the ones she had already visited, she can also modify the query she asks of the know-it-all. A know-it-all that had domain knowledge of bakeries, and who knew that people have different preferences, may immediately have been able to elicit the relevant information from the bakery seeker, in order to immediately direct her to the bakery most likely to give satisfaction.

3.2 Towards Adaptive Hypertext Systems

A hypertext system allows a user to follow links from one document to another, and an information retrieval system allows a user to locate documents by describing their content. A hypertext system is not usually designed to be searched, and an information retrieval system is not usually designed to be browsed through, because the representation required of the information contained within each system does not always directly support the other mode of interaction. There is no requirement for documents to be linked to support information retrieval, just as there is no requirement for a document base to be represented by an inverted index to support browsing. However, it is of undoubted benefit that hypertext systems and information retrieval systems are both searchable and browsable.



The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential as a forum for information, commerce, communication, and collective understanding. Find out more about W3C.

Modularization of XHTML Becomes Candidate Recommendation

21 October 2000. W3C is pleased to announce that Modularization of XHTML has become a Candidate Recommendation. This period is a call for implementation, testimonials, and review by W3C Members and the public, and lasts through 17 November. The specification subsets XHTML and extends XHTML's reach onto emerging platforms like mobile devices, television, and appliances. Read more on the HTML home page.

User Agent Accessibility Guidelines Last Call Working Draft Published

23 October 2000. The W3C Web Accessibility Initiative (WAI) has issued a Last Call Working Draft of User Agent Accessibility Guidelines 1.0 and its companion Working Draft, Techniques for User Agent Accessibility Guidelines 1.0. They provide guidelines for designing user agents that lower barriers to Web accessibility for people with disabilities, and help developers create more usable software for all Web users. Please send comments to w3c-wai-ua@w3.org (archive) by 13 November.

P3P 1.0 Last Call Working Draft Published

18 October 2000. The P3P Specification Working Group has issued a Platform for Privacy Preferences (P3P) 1.0 Last Call Working Draft. P3P enables Web sites to express their privacy practices in a standard format. P3P user agents will inform users of site practices. Please send comments to www-p3p-public-comments@w3.org (archive) by 31 October.

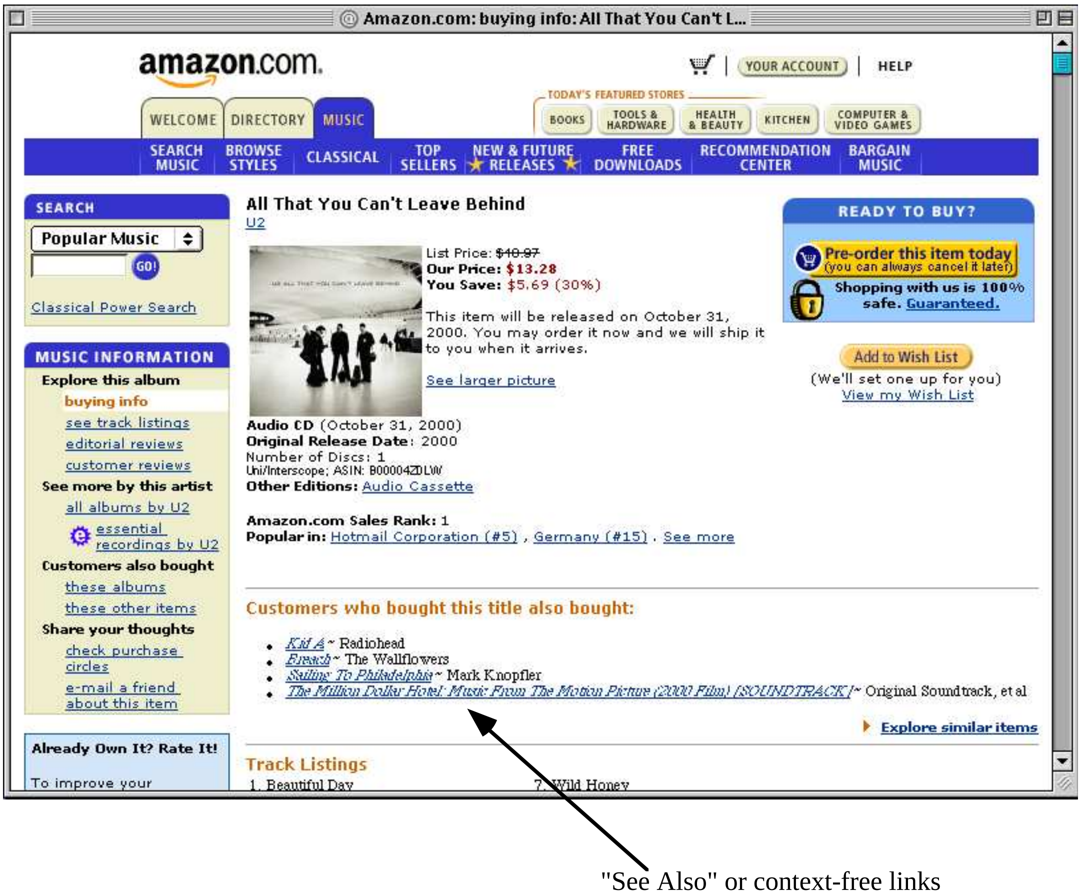
XSL Working Draft Released

Figure 3.2: Examples of navigational, index, and context-sensitive links

A hypertext system may, in fact, be considered a special-case information retrieval system. A link may be created between two documents because the link's creator believes that the destination document is the one which most suitably expands upon some information given in the source document. We can almost imagine a hypertext author posing a query and selecting the most pertinent document to link to, based on his or her knowledge of the documents available, and the requirement that the information in the destination document should fulfil. It must be stressed that there are usually no hard and fast semantics governing the use of links (but see [47] and [6] for examples of hypertext systems that use link semantics, and [94] for a discussion of link semantics in semantic networks).

According to the way links are used in untyped hypertext systems, links generally fall into the following categories: context-sensitive, context-free, navigational (guidance), and index links [15] (see figures 3.2 and 3.3). Navigational links permit the user to rapidly relocate to significant landmarks in the hypertext (to a root node, or to a node which gives an overview of the information in this part of the hypertext) and to allow movement through the hypertext following a trail laid by the author (using "Next" and "Previous" links). Index links are available through site "overviews", such as contents pages and maps. Context-free links are also lists of links, although dissimilar from index links in that they refer to documents which are relevant to the information given on the page ("See Also" links). Context-sensitive links are embedded within sentences.

Context-sensitive and context-free links are particularly interesting, because the document's author needs to make the decision that links are required in the first place, and, for each link, that the destination document is, in fact, the most suitable document to link to. In both cases, there may be more relevant documents in existence than those to which the author has provided links. The author, however, provides a link to just one document in the case of a context-sensitive link (unless the link is multi-headed), and to a number of relevant documents in the case of context-free links. The author must also ensure that the number of links presented do not overwhelm the user with choice.



All That You Can't Leave Behind
U2

List Price: ~~$18.97~~
Our Price: $13.28
You Save: $5.69 (30%)

This item will be released on October 31, 2000. You may order it now and we will ship it to you when it arrives.

Audio CD (October 31, 2000)
Original Release Date: 2000
Number of Discs: 1
Uni/Interscope; ASIN: B00004ZDLW
Other Editions: Audio Cassette

Amazon.com Sales Rank: 1
Popular in: Hotmail Corporation (#5) , Germany (#15) , See more

Shopping with us is 100% safe. Guaranteed.

(We'll set one up for you)

Customers who bought this title also bought:

- *Kid A* ~ Radiohead
- *Breach* ~ The Wallflowers
- *Sailing To Philadelphia* ~ Mark Knopfler
- *The Million Dollar Hotel: Music From The Motion Picture (2000 Film) [SOUNDTRACK]* ~ Original Soundtrack, et al

Already Own It? Rate It!

To improve your

Track Listings

1. Beautiful Day
7. Wild Honey

"See Also" or context-free links

Figure 3.3: Examples of context-free or "See Also" links

Information retrieval allows users to select from the set of relevant documents the ones which, subjectively, are most suitable. Hypertexts allow authors to give subjective recommendations to users. If we were to merge information retrieval systems with hypertext systems, then perhaps, rather than specifying the document at the link destination, the author could describe, as a query, the document which best satisfies her requirements. When a user clicks on a link, the associated query is submitted to an information retrieval system which selects the most relevant document as the link destination. To support this method of interaction, documents will also need to be represented by an inverted index. A link would allow users to navigate from one document to another by triggering a search for the most relevant document. However, although authors are able to describe the information at the destination of a link, two important elements of hypertext and information retrieval systems appear to be waylaid when information retrieval and hypertext systems are merged in this way: the reassurance that a rational entity has gone through the trouble of selecting the best document to link to (hypertext), and the ability for the user to decide which document, from a set of selected documents, is most suitable (Information Retrieval).

Adaptive hypertext systems allow individual users to view the information space from a personal perspective. The purpose of adaptive hypertext systems is to provide dynamic assistance to individual users in their search for information. This support can take various forms. Consider a document which is to be read by people with different levels of expertise in the topic under discussion. The intention is that all readers, regardless of their levels of expertise, will find the document useful. It may be necessary for the less experienced reader to have certain concepts explained in some detail. An expert, however, may soon decide that such a document is pitched at too low a level for her, because she finds the over-long explanations too distracting. If it is possible to determine the level of expertise of the reader, then it may be possible to expand on topics only if the reader requires it, otherwise keeping the discussion succinct to be attractive to the expert.

If our bakery seekers do not know their way about town as well as each other, then the local know-it-all will probably need to provide different types of information, and landmarks, to the people needing directions to the bakeries. As the town-dwellers are asking for bakeries in the first place, we can assume that they do not know the exact location of any bakery. If Tom, one of the bakery seekers, is newly arrived in town, he would probably appreciate directions to the bakery which do not presume knowledge of the town. On the other hand, if Karen is familiar with the important landmarks around town, then she would probably prefer the direction to the bakeries to be given in relation to those landmarks. If the local know-it-all does not take the time and trouble to determine the level of knowledge Tom and Karen have, then both bakery seekers are likely to be frustrated - one receiving too little information with which to find the bakery, and the other having to convert the information given to the level of abstraction required. For example, the know-it-all may describe a landmark as "the church with a spire", whereas Karen may already know the church by name.

Adaptive hypertext systems can provide support to users by reducing the size of the perceived hyperspace by filtering out information (both in terms of documents and links to documents) which are considered to be irrelevant to the user's task. In addition, the content of information in the hyperspace can be presented at the level of abstraction required by users. An important aspect of adaptive hypertext systems is gathering information about the user to accurately determine the user's interests and the level of detail required to satisfy the user.

A user model contains pertinent information about the user. This information is applied to the hypertext structure and the information contained in the hypertext to adapt both to the requirements of each user.

> by adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. [15], pg. 2.

Adaptive hypertext systems have their roots in Intelligent Tutoring Systems. Intelligent Tutoring Systems (ITSs) are able to lead students of varying skills through learning material which is dynamically tailored to the needs and requirements of the individual students. This is achieved by the ITS applying the knowledge of how to teach and its knowledge of the domain (the *domain model*) to what the student already knows (the *user model*) [3].

When the concepts behind Intelligent Tutoring Systems are applied to general hypertext systems to obtain heterogeneous Adaptive Hypertext Systems, the user and domain models tend to be more loosely related. A user model derived from a user's interaction with a domain-specific part of the hypertext will not necessarily be usable when applied to another domain within the same hypertext (just as a user model derived from an ITS teaching the C programming language will not necessarily apply to an ITS teaching diplomacy). Also, although the goals and tasks of an ITS user may be highly specified, they are less likely to be specifiable, or even automatically identifiable, in a heterogeneous environment. This leads to the question of whether a hypertext containing heterogeneous information should be partitioned into domain-specific units, with established connections between the units. This would, almost invariably, lead to rigid tree-like canonical hypertexts which provide more and more specific information about a domain as the user delves deeper and deeper among the lower reaches of the tree. Adding information to the hypertext would involve determining the correct substructure within which to place the new document. Part of the beauty of general-purpose hypertexts, however, is the freedom of association between information. In a canonical, hierarchical system, loosely related information will always be separated by a great distance (in terms of the number of intervening branches). However, in a flexible hypertext, loosely related information can be just one link away, and it is this flexibility which offers potential for information discovery. We want to be able to provide users with just the right level of automated support to locate the information they require without enforcing strict controls on how authors choose to organise and link the information they provide. The remainder of this chapter describes and discusses adaptive hypertext systems which support users with disparate needs and requirements, searching for information in a loosely organised, loosely controlled hypertext, such as the World Wide Web.

3.3 Web-based Adaptive Hypertext Systems

Brusilovsky ([15], [14]) has written excellent and in-depth reviews of the state-of-the-art in the field of adaptive hypertext systems. Instead of providing a similar analysis of the field, we have chosen to focus on adaptive hypertext systems which are Web-based, or at least accessible over the Web. Some AHSs have been designed for narrow, domain-specific, and, sometimes, task-specific environments. Although some of those AHSs will be discussed here, the research presented in this thesis is oriented towards a domain-independent, task-independent, Web-accessible environment, and so the literature review reflects that orientation.

The World Wide Web is a loosely connected network of documents which is specified by a document mark-up language (HTML, the HyperText Markup Language) and a communications protocol (HTTP, the Hypertext Transfer Protocol). HTML comprises sets of *tags* which, when surrounding ASCII text, causes a Web browser (such as Mosaic, Internet Explorer, Netscape Communicator) to interpret the text in a particular way. For example, the tags <B> and </B> surrounding the text "hello, world", will, when interpreted, cause "hello, world" to be displayed in **bold** style on a user display. HTML also includes tags for specifying link anchors and document destinations (hypertext references), which when followed (clicked on) by a user through a Web browser will cause the document identified as the link destination to be displayed on the user's screen. HTTP is the communications protocol which allows requests for documents to be made, and enables documents to be transported to the requester.

Web documents are uniquely identified by a Uniform Resource Locator (URL). The URL contains the protocol to use to effect the conversation; the identity of the host server; and the identity of the document requested. If the communications port the server is listening on is not port 80, then the port number can also be included in the URL. The URL http://www.cs.um.edu.mt/Faculty/cstaff/index.html identifies the document index.html in the directory /Faculty/cstaff on the Web server www.cs.um.edu.mt. The protocol to use to request the document is HTTP. The protocol is required because URLs can also be used to specify documents which reside on FTP, Gopher, and other servers.

Information about link anchors (the visible object which indicates a link, such as text) and link destinations are embedded within the HTML documents themselves. This implies that once an HTML document has been created it is not normally possible to modify link, link destination, and information content. However, an essential feature of adaptive hypertext systems is that information content, links, and even the destination of links must be modifiable in order to supply individual users with adapted documents. Web browsers and servers are generally flexible enough so that HTML documents can be created dynamically. A link anchor may contain a URL which is directed to a Common Gateway Interface (CGI) script which generates an HTML document on the fly, based on variable information. As the HTML document requested is generated at the time of the request, then, at the expense of some slight delay in response times, an HTML document can be individually created for different users with different requirements.

This section discusses Web-based adaptive hypertext systems, which implies that document currency is HTML, or a variant, and that HTTP is the protocol used to make requests for documents. Usually, however, additional protocols and extensions to HTML are used to give the Web an adaptive flavour, but these are more or less seamlessly integrated into the user's browsing environment.

Adaptive hypertext systems attempt to provide users with individual support by estimating users’ needs and requirements to adapt the links and information content of the documents in the hypertext [16]. By adapting the links from a document, the chances that the links will lead to relevant information are increased, and by adapting the content of objects the chances of being exposed to irrelevant information are decreased.

In [81], four different types of systems are identified which benefit from adaptivity:

- systems used by users with different requirements;
- systems used by a user with changing requirements;
- systems used by a user working in a changing environment;
- systems used by a user working in different system environments.

Most adaptive hypertext systems address the first two types of user, although the World Wide Web is also used by the latter two types of user. Detecting that a user's requirements are changing is not a straightforward task. A user's requirements can change rapidly over a short period of time, as well as slowly over a longer period of time. An Intelligent Tutoring System not only needs to identify a student's initial skills and knowledge, but needs to anticipate that these will change over time - partly because the student is learning while using the ITS, but also because the student may learn new skills and knowledge from sources external to the ITS. If a student's skills and knowledge increase dramatically between one ITS session and the next, but the ITS models only the skills and knowledge the student had that last time she used the system and does not allow for the student to learn from external sources, then there will be a significant mismatch between what the student is ready to learn, and what the system wants to teach.

A user's requirements may also change rapidly during a single hypertext browsing session, or during an interaction with an information retrieval system. A user may begin navigating through a hypertext with a specific goal in mind, but may be end up pursuing information only loosely related to the original goal.

3.3.1 Systems review

AVANTI [35] provides information about a metropolitan area to users with different requirements (for example, tour operators and tourists), different knowledge of the domain, and different abilities (for example, normally abled, blind, and physically disabled users). AVANTI is able to provide appropriate information even if not specifically asked (if it knows the user is wheel-chair bound, then it will provide information about building accessibility, for instance), and is able to make assumptions about the level of detail of explanation required by its users, as well as deciding whether incidental information is likely to be of interest. AVANTI uses an extension to HTML called Information Resource Control Structure (IRCS) to mark up documents. IRCS is used to express optional and alternative elements in a document. Information from the user model and a model of content-related multimedia objects are used to determine how the document should be adapted. Although the user model is initially based on a stereotype, User Model Construction Rules embedded in the IRCS document may be used to update the user model as the user progresses through the hyperspace.

WebWatcher [50], [1] interactively determines a user's interests and guides the user through an arbitrarily connected hyperspace, such as the Web, using information based on previous users' paths of traversal through that space. At the end of a session, a user specifies whether her information requirement has been satisfied. If it has, then links that the user followed are marked up to contain the user's search terms. The information need of the next user is compared with the user interests of satisfied users stored on each link. The most promising links are recommended, although the user is free to select from any of the available links on the page. In addition, WebWatcher can summarise all the information accessible from each link in a document and compare the user's interests with the summary to determine which links to recommend. WebWatcher can locate pages similar to the one the user is currently visiting. Unlike information retrieval systems, WebWatcher does not require term matches between a user query and a document representation. If a previous user has associated the terms "intelligent agents" with a document that does not explicitly refer to those terms, then a future user looking for information about "intelligent agents" will be directed to a link which may ultimately lead to the relevant document. Similarly, Mathé and Chen [61] describe how a compositional relevance network can be used to associate query terms with related documents. Unlike

WebWatcher, which associates queries with existing links in the hypertext, Mathé and Chen dynamically create hyperlinks between nodes based on the relevance network. In WebWatcher, a well trodden path will become more likely to be recommended to future users with similar queries, whereas in Mathé and Chen, a well trodden path will result in remotely linked nodes (the nodes at the start and end of a well trodden path) becoming directly linked. Bollen and Heylighen [8] provide evidence that a hypertext which organises itself according to its users' knowledge and usage patterns can improve future users' interactions with the hypertext.

Yan *et. al.* [95] approach the problem of directing users to relevant information from the opposite perspective chosen by WebWatcher and Mathé and Chen. Here, rather than marking up links to reflect the interests of users who have followed them, or providing short-cuts to relevant information based on well-trodden paths, user access patterns through a hypertext are analysed to classify users. Apart from improving document organisation for navigational convenience, if a visitor can be readily classified, then she can be recommended links to documents which other, similar, users have found relevant, interesting, or useful.

Kay and Kummerfeld [54] describe the Web-based provision of individualised, self-paced, teaching material for the C programming language. The student's pre-knowledge and preferences are established through dialogue prior to the student embarking on the course. The learning materials are then customised before they are presented to the student. As with AVANTI, the user model is initially a stereotype. Attribute-value pairs in the user model can subsequently be modified as the system learns more about the user, and as the student's knowledge of C increases. The C-Book tutor provides adaptive presentation support as well as adaptive navigation. HTML documents containing the teaching material are generated on the fly, so apart from customising the document content to the individual requirements, it is also possible to decide whether to present a link to the user, and if so, to which destination the link should point.

AHA [28], the Adaptive Hypermedia Architecture, is an adaptive hypermedia system development environment which evolved from an ITS. Authors can use the tool to develop generic adaptive hypermedia systems. Like Kay and Kummerfeld, HTML documents contain conditional elements whose presentation is controlled by concept-value pairs, which constitute the user model. AHA supports adaptive presentation and adaptive navigation by surrounding chunks of text, and hypertext references, with Boolean conditionals. The user model can be updated by setting or updating concept-value pairs within the HTML document. The onus is on the adaptive hypertext author to identify the possible user groups which will visit the hypertext, provide domain knowledge which is also recorded as concept-value pairs, and to control how and when information is presented to each user.

ELM-ART [17] is another example of a programming language tutoring system, this time for Lisp, delivered over the WWW. As with Kay and Kummerfeld, KN-AHS [56], and to an extent AVANTI, ELM-ART requires domain knowledge in order to know how concepts and learning material relate to each other. ELM-ART can determine what students are ready to learn or whether the concepts on a particular page require as yet unlearned prerequisite knowledge, and links are annotated accordingly. Once again, pages are constructed dynamically when requested by a user. An overlay user model, which directly overlaps with the domain model, records the concepts known by each student and the skills level of the student for each concept. The domain model records, for each page, the skills and concept knowledge required by the student in order to assimilate the information on the page. The content of the page is modified, prior to presentation, to reflect the student's educational state. Once the prerequisites have been obtained, and the newly presented material is properly understood, the user model can be updated to reflect the change in the student's knowledge.

InterBook [16] is a general extension to ELM-ART. Whereas ELM-ART was designed around the domain of the Lisp programming language, InterBook can be used to model any subject domain, although the underlying assumption is that the domain will be educational. InterBook was designed to investigate the usefulness of adaptive navigation support, specifically through the use of adaptive link annotation. Link annotation gives users more information about the current state of nodes behind the annotated links. In particular, users can tell at a glance whether the node contains information which is ready to be learned, not ready to be learned (perhaps because the user requires knowledge of not yet learned prerequisites), or whether the student already knows the information in the node. InterBook is highly dependent on knowledge of the concepts which comprise the domain, and the relationship between concepts. The user model, which is initially instantiated as a stereotype, is updated as the students navigate through the information space.

With the notable exceptions of WebWatcher, Mathé and Chen, ELM-ART and InterBook, and Yan *et. al*., the adaptation rules and user model update rules are embedded within the HTML documents themselves. When a document is requested by a user, it is pre-processed to activate the sections, links and link destinations which are consistent with the current state of the user model. If appropriate, the user model can be updated to indicate that the student's knowledge of the concept associated with the document has changed. ELM-ART and InterBook rely on annotated HTML files to contain information, with the domain knowledge of concepts and their relationships represented separately. Mathé and Chen use a relevance network which captures associations between user queries and the documents judged, by users, to be relevant to their query. Yan *et. al.* use a vector space model to cluster user browsing patterns. During an on-going user session, the partial session is compared to the vector space to identify similar sessions, so that links to possibly interesting information can be dynamically recommended. WebWatcher also uses a vector space model to represent link descriptions and against which to compare a user's declared interests.

Even more notably, only WebWatcher, Mathé and Chen, and Yan *et. al.* modify the hyperspace as a direct result of user interaction. Mathé and Chen's relevance network can be manipulated independently of the information space it represents to allow, for example, two documents to become more closely located in the hyperspace than they were originally, to reflect user access patterns. In Yan *et. al.*, the current user's browsing pattern will ultimately have an effect on recommendations of links to future users. In WebWatcher, vectors are updated whenever a user follows a link which leads to information relevant to her declared interests. From Bollen and Heylighen [8] it can be expected that the usage patterns may stabilise after some time, until trends cause the usage patterns to change again.

WebWatcher, Mathé and Chen, Yan *et. al.*, and Bollen and Heylighen all employ clustering strategies. In Information Retrieval, documents cluster when they are similar to each other, so to make documents cluster, their representations can be modified to make them similar to each other. In a hypertext, documents get clustered by linking. In user modelling, users get clustered if they exhibit the same characteristics as the stereotype. To provide adaptive support, we can either categorise users, and then assume that the same class of users is interested in the same documents, or we can categorise information, and assume that if a user is interested in one document, she will also be interested in other documents containing information from the same category. Documents can be classified according to the information they contain, and also because documents are judged by a user to be relevant to the same declared user interest (the search query). Classifying documents can be very closely related to classifying users - it just depends on the perspective. For instance, let's say user A chooses items **a**, **b**, **c**, and **d**. Another user, B, comes along and chooses items **b** and **c**. We can observe the problem from two perspectives. We can either say that user B is behaving in a similar way to user A, and so we should assist B by giving B what A took, or else we can look at it from an information perspective and say that usually when **b** and **c** are taken, **a** and **d** are taken too. In either case, user B will end up with the same set of items. Which solution is "better" depends on which is more reliable. If the system is document-centred then how users are supported depends on how the document base changes over time. If the document base never changes, then users with the same stated interests will always be given the same set of documents. If the system is user-centred, then the documents given to a user based on her stated interests will depend largely on how prior users with similar interests have influenced the system.

El-Beltagy *et. al.* [33] describe QuIC, a Web-based environment in which browsing through information and searching for relevant information are mutually supporting tasks. Like HyperContext, QuIC assists future users by using information about documents that previous users have found useful. QuIC can take an existing and known-to-be useful link between a source document *S* and a destination document *D* and re-use the link in some other document *X* which is similar to document *S*. When a future user visits document *X* a link to document *D* will be available, even though the author of document *X* had not constructed that link. When a user's interests are known to QuIC, links can be made available only if they are likely to be relevant to the user's interests.

A benefit of true self-organising hypertexts is that hypertext authors need not be too concerned with ensuring that they have provided all the information that might ever be needed, because over time, a self-organising hypertext will adapt itself to patterns of use. This is a major difference between a general-purpose AHS and an ITS. An ITS assumes, quite correctly, that it is the source of knowledge, and changing its organisation could have a detrimental effect on the ability for students to learn under its guidance. With general-purpose information spaces navigable through a hypertext browser, however, the information consumer provides the market forces that determine which information, and which information sources, will survive, and how and when that information gets used. Crucially, it is quite clear that unless the adaptation rules and domain representation are kept separate from the information in the hyperspace it is not possible to provide support for users to participate in, or contribute towards, the evolution of the adaptive hypertext.

While discussing SNITCH [62], a model of adaptive hypertext based on semantic networks, Mayfield remarks that 'it seems... crucial in most contexts for the system to adapt to a user within a single session'. From the discussion of adaptive hypertext systems presented in this chapter, it is clear that whereas ITSs must be able to adapt to the precise needs and requirements of individual students before a session can even be started, domain independent AHSs can require a period of intensive use (training) before they can adapt with any accuracy to their individual users (because the recommendations made are based on decisions made by prior users). The major reason for this discrepancy appears to be the lack of a domain model to which a user model can relate. As an ITS's domain model is constructed by an expert, it does not need to be corroborated by future users. In general-purpose AHSs which lack a domain model, it is user behaviour over time which corroborates the hyperspace's organisation. However, it remains essential that domain independent AHSs can adapt to their users within a single session, because it is during this time that a user requires most support. As time goes by, the user will either find the required information without support, give up, or else will probably no longer be interested in the same information. Although AHSs can help with long-term user interests, they are most needed for short-term user interests. If a user has a long-term interest, then the user is likely to be fairly knowledgeable about the domain, and consequently is likely to be able to express queries which will accurately describe the information need. With short-term interests, the user is more likely to be unfamiliar with the domain, and so requires more assistance to construct queries and navigate through unfamiliar hyperspaces. An AHS deployed in a heterogeneous environment should replace the requirement for the user to develop a mental model of the domain, if the domain contains information in which the user has only a transitory interest, but should assist the user in constructing such a model of the domain if the user has a long-term interest in the information.

3.4 Reflecting short-term interests in a user model

Foltz and Dumais [36] discovered that generating a user profile by extracting features from documents described as relevant by users yields better results than using only an unmodified sets of terms given by users to describe their own interests. This is consistent with the use of relevance feedback in general information retrieval systems. In Information Retrieval (IR), a user initially describes her information requirement. The IR system searches for relevant documents and presents these to the user. Over a number of iterations, she will identify one or more of those documents which are subjectively considered relevant, and perhaps identify some as irrelevant. The IR system will then modify the user's original query, normally using the Rocchio method of query reformulation [74], or a modification of it (for example, [86], [18]), to improve the original query to increase the likelihood that the relevant documents will be retrieved, while reducing the likelihood of retrieving those marked as irrelevant. In so doing, other previously disregarded documents may be relevant to the modified query. The relevance feedback loop continues until the user finds the document she is looking for, or else gives up. The most obvious reasons for relevance feedback improving the subjective quality of relevant documents are that the user is not adequately familiar with the topic to use the appropriate terminology in her original query, or that she does not use the terms commonly used in the document collection to express her query.

Imagine that Tom, the intrepid bakery seeker who is newly arrived in town, uses the town's on-line information services system to find information about a "bread shop", rather than a "bakery". The information system contains collections of documents related to the town, such as facilities, retail outlets, entertainment spots, restaurants, and so on. These documents are in the form of short descriptions of products and services, provided by the suppliers themselves. Normally, an IR system will return as relevant those documents which contain the terms specified in the query. This places an onus on the information providers to anticipate the possible terms to which their offering could be considered relevant, so that the terms can be included in the description. In our example, only one of the bakeries has included "bread shop" in the description, and this document only is presented to Tom. If Tom instructs the IR system to locate additional similar documents, then the relevance feedback system will modify Tom's original query to include the terms in the relevant document which are also considered to be suitable descriptors. If one of those terms is "bakery", then the IR system will be able to retrieve other relevant documents for Tom. Although the information Tom seeks is not directly present in the form in which he describes it, the IR system is able to extract the actual features that have been used to index documents relevant to Tom's request, based on his instruction to find documents similar to the only document originally found to be relevant.

One of the more prevalent problems with automatically generating a user model to reflect a user's short-term interests is detecting that the topic of interest has changed [58]. This has a significant bearing on the accuracy of the (short-term) user model because if a user moves rapidly from one interest to another, the user model representing the new interest should not be effected by the previous interest. For example, during a conversation between two people, the topic of conversation can change, gradually or instantaneously. It would be quite confusing if an anaphoric reference was incorrectly bound to a subject which was part of an earlier topic. Likewise, if an IR user originally sought information about Ferrari sports cars, which are manufactured in Italy, and then turned her attention to Parmesan cheese, it would normally be quite incorrect to draw the conclusion that she is interested in fast dairy products!

Educational AHSs are able to detect that a goal or task has been accomplished because the information is elicited via a test of the student's knowledge or skills. If the student has passed the test, then the student's goal or task has been achieved and she is able to progress to the next milestone. If the task is an information seeking task using an IR system, then, usually, only the user is in a position to indicate whether the task has been completed, successfully or otherwise. In a general-purpose hypertext system the user does not even need to state her information requirement, let alone indicate to the system when her task has been completed. WebWatcher uses IR techniques to support browsing.

It requires the user to specify her initial requirement, and subsequently guides the user to information which others previously looking for similar information have found relevant. The user marks the end of the session by explicitly informing WebWatcher at the same time as giving a relevance judgement. WebWatcher associates the path taken to reach the final document with the user's stated interests. Yan *et. al.* [95] do not require the user to specify his interest. Instead, the path the user has followed so far is compared to paths followed by previous users, so that based on usage patterns, links to interesting documents can be recommended.

3.5 Summary

We have identified that Adaptive Hypertext Systems fall into two broad categories: domain-specific educational systems built around hypertext, and heterogeneous adaptive hypertext systems. Educational systems normally accurately model the domain and a user's knowledge, and the primary goal of such systems is to assist the user attain greater knowledge of the subject matter. Heterogeneous adaptive hypertext systems generally cannot accurately model the domain (except at great expense). A user's interest can be elicited through a pre-stated information need, or else the user can be stereotyped according to access patterns through hyperspace. The user interest is normally represented as a user model.

A user's interest may be long-term or short-term. The user is likely to be familiar with the domain (in terms of information organisation and terms of reference) if her interest is long-term. Users' short-term interests tend to be harder to ascertain, because users are unlikely to be familiar with the domain, but this is precisely where heterogeneous adaptive hypertext systems need to provide support. An effective solution is also required to detect that a user's interest has changed, possibly within the same browsing session, otherwise the hypertext will be unable to provide the user with relevant information.

If a user is unfamiliar with the *lingua franca* of the domain, relevance feedback in information retrieval systems can enable the user's original query to be automatically reformulated based on user provided relevance judgements on retrieved documents. This method can be effective in both domain-specific and heterogeneous collections of information.

Educational hypertexts are unlikely to be social hypertexts in the sense that users actively participate in the organisation of information, because the structure must be maintained to teach effectively. Heterogeneous hypertexts are social because user activity can, directly or indirectly, effect the organisation to enable the hypertext to provide better assistance to future users.