Chapter 10

Conclusion and Future Work

10.1 Introduction

Our journey began with a vision of a social hypertext which could be adapted according to the needs of its community of users, and which could adapt to the needs of its individual users. The HyperContext framework which we propose permits this vision to be achieved. The important adaptive features from the framework are implemented as a prototype. We show that the following mechanisms provide adaptive support to users as they browse through hyperspace:

- multiple interpretations of information in context;
- distinguishing between contextual and superficial relevance of information in Information Retrieval-in-Context and Adaptive Information Discovery;
- the derivation of a short-term user model based on salient interpretations of nodes accessed in a context session to capture a user's short-term interest;
- the automatic generation of a search query from the user model to recommend relevant information through Adaptive Information Discovery;
- and, guiding a user to relevant information by link recommendation.

This chapter summarises the main proposals and findings of the thesis (Section 10.2), and recommends further work on the HyperContext prototype to more fully realise the vision (Section 10.3.1). The HyperContext framework itself can also be extended (Section 10.3.2). Finally, there is considerable scope for future research, both within HyperContext (Section 10.3.3), and in other domains as a result of HyperContext (Section 10.3.4).

10.2 Summary of main proposals and findings

The crux of the thesis revolves around the use of multiple *interpretations* of information to provide a framework for adaptable and adaptive hypertext. In any hypertext, each piece of information is normally juxtaposed by other information via links. Two or more hypertext users may encounter the same document although they may have followed different paths to reach it. Those two users may well describe different aspects of the document as relevant to their needs and requirements. The HyperContext framework allows users to create different interpretations of information in context, which will be available to future users.

Interpretations permit a hyperspace to be partitioned as it is traversed. Out-links are associated with interpretations, so the same document in different contexts can have different out-links, or common out-links which have different destinations in each interpretation. This gives us the notion of a *context path*, a sequence of interpretations linked in some context. As a user browses through hyperspace, a node can be *contextually relevant* if it is on the same context path as the user and relevant to a user's interests. Other documents in the hypertext which are relevant to a user are *superficially relevant*. While a user browses through a hyperspace, the interpreted documents accessed on the path of traversal form the user's *context session*. A user may make superficially relevant information contextually relevant by extending a link to it from a node in the context session.

Users searching for information are supported by three Information Retrieval mechanisms: *Traditional Information Retrieval* (TIR), *Information Retrieval-in-Context* (IRC), and *Adaptive Information Discovery* (AID). TIR enables a user to specify a query and presents the user with all relevant interpretations, regardless of context. IRC presents contextually relevant interpretations in response to a user supplied query. AID utilises a *short-term user model* to assimilate a user's short-term interest, based on the context session, and can automatically generate a query on behalf of the user. Superficially relevant information is recommended by AID to the user via "See Also" links. If, following a search, a user hyperleaps to a node containing superficially relevant information she is given the option to make it contextually relevant by extending a link to it from within the current context session, otherwise a new context session is initiated. On the other hand, HyperContext can guide the user to contextually relevant information by recommending links through the hyperspace.

The short-term user model is initialised at the beginning of a new context session. We distinguish between a long-term and a short-term interest. A long-term interest is one which persists across many context sessions, and perhaps lasts for weeks, months, or even years. A short-term interest is transient. It may extend over a small number of context sessions, but it is unusual for it to last for long, although short-term interests can develop into long-term interests. We express the user's short-term interest as a function of the interpretation of documents that the user has seen in the current context session. The user's perceived interest in the current document is represented as a *salient interpretation*. Salient interpretations are combined in the user model according to a weighted *scale of confidence* in the salient interpretation's usefulness in identifying a relevant document.

We implemented a prototype HyperContext client and server to demonstrate the adaptive aspects of the framework. We automatically converted a WWW site consisting of 170 nodes into a HyperContext hypertext, with the intention of conducting empirical studies to evaluate our approach to adaptive hypertext. The implementation approach precluded the prototype's use as a multi-user test-bed. However, we were able to automatically evaluate the manner in which the salient interpretation is derived, and the advantages of multiple interpretations of documents over single re-usable document descriptions, even though the interpretations were system generated, rather than being generated by "real" users. Finally, we conducted user-based experiments to evaluate AID's ability to recommend a relevant document based on the short-term user model.

The HyperContext framework permits a hyperspace to be adapted by its community of users to reflect how the information is actually consumed. Users are able to reuse information, regardless of who authored or "owns" the information, by creating new links between existing nodes. Users are also able to describe the information in the destination node which is relevant to them, to provide an interpretation of the information in the node. Each interpretation of a node is represented by a vector of weighted terms. The parent node containing the link source anchor and the link itself provide the context in which the destination node will be interpreted whenever it is accessed via that link. The interpretations of a node collectively reflect the different ways of describing the information contained in the node. In an Information Retrieval (IR) system, a document is normally represented by a single vector of weighted terms, which may then be used to attempt to satisfy a large number of different requirements. Although users are sometimes able to give relevance feedback, this information is normally used to modify the user's query, rather than to modify the document's vector representation. As a result, an IR system used by another user with an identical information need normally cannot take advantage of its previous users' experience to improve its quality of service, whereas a

HyperContext hypertext can as its users record how the information they access is relevant to them.

The interpretations of information are searchable and retrievable through an interface between the HyperContext framework and an external information retrieval system. The HyperContext prototype interfaces with SWISH-E, which provides external information indexing and retrieval services to HyperContext. When a user searches for information, HyperContext invokes the external IR system and retrieves *interpretations* of documents which are relevant to the query. Depending on which search mechanism the user invoked, HyperContext will either present the user with a ranked list of relevant interpretations (TIR), or it will guide the user to a contextually relevant interpretation by recommending links to follow along a context path (IRC). Non-adaptive hypertexts normally cannot guide users to information without the hypertext author having first created a purpose-built trail. On the other hand, adaptive hypertext systems can guide users to relevant information using trails or paths of traversal frequently travelled by previous users. However, in HyperContext we distinguish between contextual relevance and superficial relevance to guide users to relevant information along a context path which other users have previously created.

A benefit of adaptive hypertext systems is that they are able to automatically or semi-automatically determine a user's interests. In HyperContext we distinguish between a user's short-term interest and her long-term interest. We assume that a user is likely to require greater support in her search for information to satisfy a short-term interest, because she is likely to be unable to accurately represent her information need. We must also detect when the topic of a user's short-term interest has changed, otherwise our representation of the user's interest may be contaminated by no longer relevant information. We also distinguish between domain-specific adaptive hypertext systems, such as AHS-based Intelligent Tutoring Systems, and general-purpose AHSs. We cannot, without incurring great expense, accurately model the user and the domain in general-purpose AHSs. In HyperContext, we monitor the scope of a user's short-term interest through a construct named the context session. As long as the user is traversing through hyperspace within the same context session, and has not hyper-leaped out of it, we assume that she is still searching for the same information, and that she has not yet located it.

We construct a model of the user's short-term interest based on the interpretations of nodes that she has accessed in the context session, assuming that each accessed interpretation in the context session is only partially relevant to her information need (otherwise she would have located the required information and terminated the context session). As an accessed interpretation is considered only partially relevant to the information need, we establish which terms are likely to be relevant and the degree to which they are relevant by deriving a *salient interpretation* of the node. The salient interpretation is derived using a modification to the Rocchio relevance feedback method, which compares the accessed interpretation of a node to all the other interpretations of the same node to establish which terms are likely to best represent the user's interest in the node. A *scale of confidence* is used to weight each salient interpretation of the interpretations accessed during the context session, to reflect HyperContext's confidence in each salient interpretation's ability to contribute information about the user's short-term interest. The weighted salient interpretations are finally combined as a model of the user's short-term interest.

The Adaptive Information Discovery (AID) search mechanism is an autonomous tool which, if active, generates a search query on the user's behalf by extracting terms from the short-term user model. The user can be guided to information that is contextually relevant as well as being presented with a list of superficially relevant "See Also" references.

The HyperContext framework is the result of research which crosses the boundaries of the domains of adaptive hypertext, hypertext, user modelling, information retrieval and context. We believe we have contributed to the area of adaptive hypertext by incorporating automatic relevance feedback mechanisms into the derivation of the model of the user's short-term interest. We also believe we have extended research into adaptive hypertext systems by incorporating explicit representations of context into hypertext systems which permits multiple interpretations of the same information to be represented and manipulated to give individual users adaptive navigation support. These conclusions are supported by the experimental results obtained from an implementation of the important adaptive features of the HyperContext framework.

10.3 Future Work

There are considerable opportunities for the continued implementation of the prototype (Section 10.3.1), further extension of the HyperContext framework for adaptable and adaptive hypertext (Section 10.3.2), and future research into the application and use of interpretations of information in context both in HyperContext (Section 10.3.3) and in other domains (Section 10.3.4).

10.3.1 Prototype implementation

The HyperContext prototype does not properly support multiple simultaneous user sessions. This has obviously had severe restrictions on the ability to test the prototype, although essential features of the framework have been evaluated. Most of the restrictions are due to the naming conventions for some temporary files created by the modified version of SWISH-E as well as the HyperContext server. In addition, some of the essential services are not all fully re-entrant, meaning that if two or more users simultaneously attempt to execute certain functions, the actual behaviour of the system will deviate from the intended behaviour.

The HyperContext framework permits users to create and modify interpretations of existing information, even when they are not the owners of the information or interpretations. This feature to support adaptable hypertext is missing from the prototype, so the prototype's hyperspace cannot be modified. Apart from supporting the physical creation of interpretations, a tool set must also support users in their selection of terms which will be labels in the interpretation, and in the selection of the destinations of links. Part of this functionality would be present in Adaptive Information Discovery (AID). If AID has helped the user locate relevant information based on the user's path of traversal through the hyperspace, then the user may decide to create a link between a document visited during the context session and the relevant document.

The unmanaged creation of interpretation and links may quickly result in an overwhelming number of interpretations of the same document, many of which may be quite similar. Apart from space management requirements and efficient methods to quickly locate specific interpretations, a tool set is required to support the deletion of interpretations and links. Identifying and resolving the issues associated with interpretation and link deletion is also a research opportunity (Section 10.3.2). Automatic link and interpretation management may employ techniques such as link and interpretation ageing or thresholding accesses to interpretations. The major issue associated with link and interpretation deletion is that unless the interpretation is a leaf node, the context paths following the interpretation targeted for deletion cannot continue to exist (primarily due to the deleted interpretation providing the context for its children).

The HyperContext framework and the prototype, although containing no examples of non-textual data, support information in multimedia formats if the interpretations are textual. The HyperContext server uses the services of external information retrieval systems, and so it should support multimedia information retrieval systems. In HyperContext, links are held separately from the document content, and are overlaid at run-time. An open standard would be required for the seamless overlay of links onto documents of any media type. The related issue of ensuring the consistency between links and the anchor text to which they are bound, even when the document content is edited, is a research opportunity. This is an issue for any hypertext system which separates links from the documents to which they relate.

The existing HyperContext prototype uses a short-term user model only, although the framework recommends the parallel use of a long-term user model, with interactions between the two user models. The long-term user model would probably need to contextualise information, as users may have several disparate long-term interests, the specific details of which may conflict with each other. The long-term user model may help the serendipitous user [22], as well as assist with the differentiation between multiple interpretations of the same document, when these are equally relevant to a user query. The long-term user model may also be employed to identify interesting incidental connections between a user's short-term interest and her long-term interests. In addition, the long-term user model would be able to identify whether the user's current context session is related to a long-term interest or not, in which case the level and type of support offered by HyperContext may change accordingly.

One of the problems with massively distributed hypertexts, such as the WWW, is the lack of a guarantee that the information contained in a newly created node will, eventually, be indexed. This is largely due to individual servers being unaware of the rest of the distributed hypertext. We advocate the use of a co-operative system of servers and indexers. Indexers must identify themselves to servers, which record the indexer's identity and time of access. Servers will notify their administrators if they have not been recently indexed. For additional assurance, a buddy system can be employed, whereby each server knows the identity of at least two other servers (with the administrators' permission). Any indexer will be given the location of the buddy servers, to ensure that these other servers will also be indexed. This is useful for Traditional Information Retrieval, and Adaptive Information Discovery. Information Retrieval-in-Context is an inherently distributed information retrieval system which uses context paths to direct search and control retrieval, so this problem is not as evident.

HyperContext is intended to be an open hypertext. Interpretations of and links to and from documents created by any application as well as universal data formats, and any media type, should be supported through appropriate interfaces.

The short-term user model is intended to not only be inspectable, but also to be modifiable. The prototype, however, supports only inspectability. Users should also be able to add terms to and delete terms from, the user model, as well as promoting and demoting terms to reflect the user's interest in them.

The short-term user model has been evaluated when the user starts a context session by directly accessing an interpretation. Users can also initiate a context session by performing a Traditional Information Retrieval search and selecting a relevant interpretation from the results. When this happens, we advocate a different method of deriving the short-term user model (Chapter 5.8.2). However, this has not been tested. A context session may also include a search in context (IRC) as well as Adaptive Information Discovery. The corresponding interactions with the user model are untested.

Although the HyperContext framework explicitly supports links with different destinations in different contexts, there are no examples of these in the hypertext converted for use with the HyperContext prototype. The research opportunity here is concerned with devising evaluation methods for a hypertext which supports the re-use of links with different destinations.

The HyperContext client has been implemented as a Java applet which is downloaded from a Web server (zeus.cs.um.edu.mt). Consequently, the HyperContext client is permitted to communicate only with the HyperContext server also resident on zeus.cs.um.edu.mt. In Chapter 7.5, we recommend that to overcome this restriction the client is split into two modules: the HyperContext Control Panel (a Java applet) and the HyperContext client (a Java application, running on the user's local computer). The Control Panel will be loaded from the user's local computer, which would therefore allow communication with the local client only. The local client, however, will be able to communicate with any HyperContext server as it is a Java application and not subject to the same security restrictions imposed on applets.

10.3.2 Extensions to the HyperContext framework

Non-linear context sessions

The framework represents a user's context session as the linear path taken from the first node the user accessed to the node the user is currently visiting. Any nodes which were visited and subsequently backtracked from are not represented. It is worth investigating how inclusion of such nodes in the context session would affect the user model.

Aggregate nodes

Context paths in HyperContext can be summarised using a method similar to the automatic generation of a user model representing a linear path of traversal. The summary could be represented as an aggregate node. The aggregate node can be used to give an indication of relevance to a search prior to the nodes in the path being queried. The aggregate node can answer questions of the form "Does the context path contain the required information?" and "What is the context path largely about?". These advantages must, however, be weighed against the overheads of maintaining aggregates, both in terms of the additional space required, and the costs of updating them in a rapidly changing environment. Aggregate nodes may be especially important for Traditional Information Retrieval to select which interpretation of a document should be offered to the user when many interpretations of the same document are equally relevant (Chapter 5.6).

Link and node typing

The HyperContext framework recognises only one link and node type. From the discussion on adaptive hypertext systems (Chapter 3), we know that there are many common link types (for example, navigational, context-sensitive, context-free and index), which are normally distinguishable by their use (the WWW, for example, does not directly support link typing, although WWW links serve a multitude of different uses). We may wish to distinguish between links that merely re-position the reader within the current document, or direct the user to important landmarks (for example, the hypertext site's root node), and links whose purpose is to lead to more specific information. This especially important if interpretations are created automatically, and for Information Retrieval-in-Context to limit search to information-bearing paths, perhaps ignoring the purely navigational ones which may simply lead to cycles and loops within the search space.

Should aggregate nodes become part of the HyperContext framework, then a HyperContext server needs to know that these exist to support automatic search, and are not be displayed to the user. Some other node types which would be useful in HyperContext are gateway nodes and composite nodes. A gateway node is an ordinary node which can be displayed to a user, but which contains a significant proportion of out-links when compared to actual content. Ordinarily, a document contains more (textual or multimedia) information than links to other documents. However, a document which contains a higher proportion of links to other documents than information acts as a gateway. Such a node is also likely to have significantly more contexts than normal. It may be, however, that it is not useful to support a different interpretation for each context. Gateway nodes may require a standard interpretation which applies to all contexts. Despite this, it may not be appropriate to generate the same salient interpretation of the gateway node for each context in which it may be accessed. It may be more practical to treat a gateway node as having no effect on the user model. The gateway node may act as a look-ahead which allows the user to make an appropriate selection of a link, without having an effect on the state of the world.

A composite node is one which is itself composed of other nodes. When a composite node is accessed, the context determines whether the other nodes should be unfolded and displayed to the user, without the user being aware that the node has been expanded. This concept is not new, and it is already used with great success in MetaDoc [9]. For example, rather than providing a link to a glossary entry for a term in an interpretation, a composite node could automatically include the linked document within the node. Composite nodes can be used to optionally include text, graphics, audio, video, and other composite nodes, given the appropriate interfaces to the respective data formats.

Adaptive Presentation techniques

Adaptive presentation techniques are a powerful mechanism for dynamically modifying the content of nodes to suit the preferences and requirements of individual users. The HyperContext framework does not make any direct recommendations for the adaptive presentation of information. Adaptive presentation can be achieved through composite nodes (see above).

Link and interpretation ageing

Link and interpretation ageing may be required to manage the number of interpretations of a node. HyperContext is a social hypertext, with its users able to create interpretations of any information, regardless of whether they are the owners of the information. There is a possibility that interpretations are created which are never again accessed. This is a waste of resources and should be minimised. Ageing links and interpretations need not necessarily result in their deletion (this has its own associated problems, see Section 10.3.1), but may instead result in their compression and archival. Research would need to identify when an interpretation and its associated links are candidates for archiving, how to reduce their consumption of resources, and how to resurrect them should they be required again.

Adaptive Information Discovery recommendations based on skim and deep reading

The short-term user model, $UM_{adaptive}$, from which a query is generated for AID is composed of salient interpretations of nodes in the context session. The salient interpretation is derived using interpretation-. For evaluation purposes, we also used a control user model, $UM_{control}$, which constructed the salient interpretations using control. During the evaluation of AID (Chapter 9.7), we noticed a marked difference between the relevance judgements given to recommended documents by users who read quickly through documents in a context session and those who may have had a deeper understanding of the same documents. Skim readers tended to award a higher relevance judgement to the document recommended by $UM_{control}$, whereas thorough readers preferred the document recommended by $UM_{adaptive}$. This observation deserves further study, as its results could have a significant impact on User Modeling, Information Retrieval and automatic relevance feedback, and adaptive systems in general.

Dynamic data

In the WWW, some information either becomes invalid, or else changes in scope, after a while. Stock market data, weather reports, and certain news items have a relatively short currency before they become historical. The HyperContext framework does not explicitly treat dynamic and static information any differently. Whereas interpretations for static and historical information are likely to be similarly constructed, rapidly changing and short-lived data may require management as new node types.

Link hiding and link disabling

To demonstrate HyperContext, we converted a Web site (www.w3.org) into a HyperContext hyperspace (Chapter 8). As a user browses through the HyperContext hyperspace, we show all links that were available in the original HTML document coloured blue and underlined, whether or not the link is accessible via an HCT button (Chapter 7.4.2).

This approach to adaptive navigation through link disabling is probably not satisfactory, for reasons explained in Chapter 7.4.2. Link hiding may prove to be a better approach. However, other research has shown that indiscriminate link hiding may lead users to build inconsistent mental models of hyperspace (for example, [68]). Further research is required to establish which of link hiding, link disabling, or link annotation is more suitable to provide an adaptive linking solution for HyperContext.

10.3.3 Other research opportunities in HyperContext

Efficient context expansion

We require an efficient algorithm for determining if two nodes are on the same context path. We need to identify whether two interpretations are contextually related in as short a time as possible. We have suggested a parallel solution, but this needs to be weighed against the cost of computational complexity and resource consumption, especially when a significant number of interpretations are relevant to a query. A related research area is extending the notion of contextual relevance to nodes which co-exist in the same context sphere, rather than limiting the relationship to nodes which co-exist on the same context path.

Using adaptive methods to decide when a user model can be trusted

The HyperContext framework and prototype use a scale of confidence to determine the point at which the short-term user model can be trusted enough to derive a query from it. There is some scope to dynamically determining after how many link traversals the user model can be trusted. The scale of confidence may be user modifiable, but perhaps an adaptive solution is possible. If HyperContext can anticipate the user's next action, based on the user model and the interpretations surrounding the node the user is reading, then HyperContext may be confident that it has adequately represented the user's interests to recommend another document with impunity. Actions that a user may perform and that may be anticipated include: following any link; following a specific link; returning to the previous node; terminating the context session; performing a search; or hyperleaping to another object (perhaps one selected from a list of favourites or bookmarks). Obviously some actions are better than others at determining the level of confidence in the user model (context session termination and hyperleaping would normally result in the context session, and consequently the short-term user model, being discarded. It would be inappropriate for HyperContext to recommend a document based on the user model just because HyperContext correctly anticipated a hyperleap).

Ontology

As with User Modeling, Information Retrieval and classification systems, there is scope to determine the extent to which an ontology will improve precision, recall, and user model accuracy. A useful research project would be to convert a thesaurus into a HyperContext hyperspace.

Object migration

Interpretations and links files normally reside on the same server as the document to which they relate. Although the document is modifiable only by the file's owner, the interpretations and links files are modifiable by any member of the HyperContext community. There is no specific reason why the document, interpretations and links files should reside on the same server, and, in some instances, it may make sense for some interpretations and links files to be moved to a different server for reasons of security, efficiency and accessibility. For example, administrators may wish to physically separate the HyperContext server from the file server, for security. Perhaps interpretations in a context path reside mainly on the same server, apart from the interpretations of one or two documents. Those interpretations can be migrated to the same server that hosts the majority of interpretations on the path, to increase interpretation accessibility. Some interpretations may be more popular than others, attracting higher rates of traffic than the server can cope with. It may be necessary to mirror HyperContext sites on faster servers, diverting traffic at times of need. Object migration may be automatic rather than manual. As interpretations know their contexts, following the movement of an object, the object's referents can be automatically updated to maintain consistency. The HyperContext framework explicitly supports object mobility and migration, but additional research would help determine if, when and how objects should be migrated.

Automatic creation of interpretations

We have discussed the possibility of automatically creating interpretations of information (Chapter 6.3.1). We have also decided that in general it is more appropriate to support human users in the creation of interpretations than to attempt automatic interpretation creation. However, the potential for a hypertext to be self-organising, not just by creating links between relevant nodes, but also by determining how to construct an interpretation of the destination document, is exciting. Within specific domains there may be clear opportunities to provide automatic interpretation creation. This will also have a significant role to play in the automatic conversion of foreign data spaces to a HyperContext hyperspace.

Identifying new linking opportunities

Creating new interpretations of information and creating new links between interpretations are related activities. In Chapter 6.3.1, we discussed the problems associated with automatic linking, just as we did with automatic creation of interpretations. However, the prospect of a hyperspace which is completely dynamically constructed at the time it is traversed is too great a lure to dismiss as untenable. Data mining techniques would probably give an insight into how such a dynamic hypertext might feasibly be achieved.

Using HyperContext with different external models of Information Retrieval

The HyperContext prototype utilises the services of an external Information Retrieval system (SWISH-E) which is an extended Boolean model of information retrieval. The framework, however, anticipates that different HyperContext servers will utilise information retrieval systems which have different underlying models of IR (for example, statistical or probabilistic). HyperContext clients and servers use a standard internal vector-based representation, exchanging data with external service providers through interfaces. Employing disparate Information Retrieval models for both distributed IR and centralised IR will require research into data fusion techniques.

Alternative representations of interpretations

The HyperContext framework and prototype represent interpretations as weighted term vectors. When the external information retrieval and long-term user modelling services use a more domain-dependent representation, information can be lost during the representation conversion process. One particularly interesting approach would be to use rules, or meta-rules, to determine how a document should be interpreted. The underlying assumption, of course, is that the representation of the document base would include domain knowledge.

HyperContext and the Semantic Web

The Semantic Web [5] is an attempt to embed within Web documents structured information which can be used by software agents and users to locate documents based on this structured information.

At a simplistic level, a Semantic Web document incorporates relational data which can then be processed by a relational database engine to return a result to a given query. At its most expressive, data in different Semantic Web documents can be related to each other to answer queries where the answer is not contained in any single document.

Consider the following fictional press release:

> "Air Mondial is pleased to announce that it will operate flights from Mondial International Airport to Pater Noster Island, starting from Friday, 17th August 2001.
>
> Flights will depart Mondial International Airport at 08:00 on Mondays, Wednesday, and Fridays, returning on Tuesdays, Thursdays, and Saturdays at 18:00. For further information call +555-999999."

A typical information retrieval system will index the occurring terms to support full text search and retrieval. Consider a user who submits a query equivalent to the natural language request "Locate documents which contain information about flights from Mondial International Airport to Pater Noster Island departing on or after Thursday". The press release will be returned to the user as relevant. However, on closer inspection we see that although the press release contains the information requested by the user, the information retrieval systems returned the document to the user *for the wrong reasons*! The vector for the document would include the terms "Thursday", "Pater", "Noster", "Island", "Mondial", "International", "Airport", "depart", and "flights". However, with a *semantic* understanding of the press release, we quickly realise that Air Mondial does not operate the requested flight on Thursdays. Therefore the document was incorrectly identified as relevant by the IR system. Luckily, however, the document contains additional information which is relevant to the user. The Semantic Web can recognise that the document is relevant to the user if it understands what the relation "after" means. From the press release, we see that there is a flight to the required destination on Fridays, which satisfies the constraint "after Thursday".

The Semantic Web is capable of using relational information spread across multiple information sources to reliably relate together concepts. For example, the user's query may be restated as "Locate documents which contain information about flights from Mondial International Airport to Main Airport departing after Thursday". If other information on the Semantic Web is able to relate "Main Airport" to "Pater Noster Island" (indicating that Main Airport is unambiguously Pater Noster Island's airport), then the user can still be presented with the press release even though "Main Airport" does not appear in it.

Using HyperContext to mimic a Semantic Web

In HyperContext, labels serve a dual purpose. Through its weight, a label indicates the relevance of a term to a document. Additionally, a label acts as a the source of a link when the label is associated with a destination document. Currently, labels are implicitly typed.

If the HyperContext framework is extended to support explicit label types, then a label may also be associated with a value or range of values (e.g., flight information: carrier(Air Mondial), origin(Mondial International Airport), destination(Pater Noster Island), depart_date(Monday), depart_time(08:00), depart_date(Wednesday), depart_time(08:00), depart_date(Friday), depart_time(08:00)); a concept (e.g., Mondial International Airport: airport); or an authoritative source (e.g., Pater Noster Island: http://www.paterNosterIsland.com/authority which contains authoritative information about Pater Noster Island). Typed labels may be used to directly provide intelligent agents with information; to relate terms to concepts (possibly through an ontology); and to provide a mechanism with which agents can locate further information from a trustworthy source.

HyperContext maintains multiple interpretations of documents in context. A document interpretation may be composed of different labels, or of differently weighted labels. In addition, link destinations of labels can change from one interpretation to another. HyperContext uses different interpretations of information to distinguish between descriptions of documents in context; to automatically determine user interests during a context session; and to lead users to relevant information along a context path. In the HyperContext equivalent of the Semantic Web, different interpretations may support the same term related to different concepts or to different authoritative sources. Some interpretations of the same information may contain attributes (labels) which are not relevant to other interpretations (and which are therefore missing or unstated).

Implementing HyperContext using XML and RDF

Annotea [52] is a distributed annotations server for HTML and XHTML documents on the WWW. Through Annotea, readers of a document can create, amend and access annotations to a document. Annotations may be created by any reader. Annotated regions in the source document are visually enhanced so that a reader can easily identify where annotations exist. The reader can request that the annotations for a region are displayed in a separate window. Annotations may include external references (links) to other sources.

Annotea stores annotations in a Resource Description Format (RDF) [13] database, and uses HTTP to post new annotations to, and to retrieve annotations from, an RDF database. Annotea uses the XML (Extensible Markup Language) Linking Language (XLink, [30]) to refer to annotations from within a source document. An XPointer [29] is associated with each annotation to indicate to which region within the source document the annotation refers.

While browsing through a HyperContext hyperspace, a user causes a document to be interpreted and presented by following a link. An interpretation consists of a vector of weighted terms which describe a document in context and a set of out-links. It may be possible to use an architecture similar to Annotea's to implement HyperContext using XML technologies. A document interpretation can be described in an RDF database. XPointers can be used to identify source anchor text in an interpretation, and XLinks can be used to bind to the destinations of links which are active in an interpretation.

An RDF database can be queried to identify documents relevant to a query. HyperContext's Adaptive Information Discovery would use the XPointer associated with a relevant interpretation and the context of the interpretation the user is currently visiting to determine whether the relevant interpretation is contextually relevant or superficially relevant to the context. Salient interpretations would be derived from the interpretations stored in an RDF database. The salient interpretations are provided to the HyperContext client so that a short-term user model may be constructed. HyperContext's Information-Retrieval-in-Context would follow XLinks from the interpretation the user is currently visiting, and retrieve the interpretations of accessible interpretations along a context path from an RDF database to locate relevant information. An RDF database would also be updated to add or modify interpretations of documents.

10.3.4 Research opportunities in other domains

Information Retrieval

We have not seen how multiple interpretations of information affect precision and recall in a pure Information Retrieval system (based on any model of information retrieval). The indications are that precision and recall will improve. We do feel that while interpretations of information may still be created by the IR system's community of users, the source which provides a context within which information can be interpreted will probably be considerably different from the source of context in hypertext. If the Information Retrieval system supports relevance feedback and query reformulation, then the interpretation which is selected at each iteration would form part of a context session. Two user models corresponding to $UM_{control}$ and $UM_{adaptive}$ can be maintained, and the reformulated query will be extracted from the appropriate user model depending on whether the user skim or deep read the documents in the context session.

Intelligent Agents

There is considerable scope both for mobile representations of long-term user interests which trawl through a HyperContext hypertext looking for interesting information, and for the use of multiple interpretations of information to support agent technology. In HyperContext, user's short-term interests which are initially unsatisfied can be given a life span during which they will report back to the user any newly created relevant interpretations.

Information re-use

The HyperContext framework is useful for any domain which requires information re-use in different contexts. Consequently, the research presented in this thesis could be useful in domains such as Case-Based Reasoning; Version Control; inter- and intra-organisational sharing of resources especially for databases and data mining; mobility, where context may be associated with physical access methods (for example, office-based, home-based, on-the-move); e-commerce and e-marketing, where context may be associated with demographic spectra; expert systems; and Intelligent Tutoring Systems and e-Learning.