Chapter 5

Adaptability and Adaptivity: Putting the user in the picture

5.1 Introduction

In Chapter 4 we discussed the structural features of the HyperContext framework that support adaptivity and adaptability. In this chapter, we describe the framework from the user's point of view. We first describe the adaptable features that allow users to add multimedia documents, and create interpretations of documents by linking to them in context. We then discuss how the HyperContext supports adaptive browsing and adaptive information retrieval, by using interpretations of information in context to derive a model of a user's short-term interest to guide the user to relevant information.

5.2 User adaptation of a HyperContext hyperspace

The Structure and Object Layers (Chapter 4.3) contain the building blocks of an adaptive HyperContext hypertext. Without the ability to create new nodes, links, and interpretations, a hypertext would be a static edifice, which, once designed and implemented by its authors, would not be able to change according to users' requirements over time.

A new document is added to a HyperContext hypertext using the Object Layer's createProfile function to build a representation, or a *profile*, of a multimedia document. The profile contains details of the multimedia document's location in the world outside HyperContext; the protocol to use to access the document; and the document's type, which specifies how HyperContext can manipulate it. A profile will also contain references which bind link source anchors to their respective regions in the multimedia documents. Existing profiles may be modified through the modifyProfile function. This function will usually be invoked only to add or modify link source anchor regions. Any user, not limited to the owner of the multimedia document, can create a profile for any document. At the same time as the profile is created, an interpretation of the document in the context bottom is created using the createInterpretation function. Without at least this interpretation the document cannot be represented in the Structure Layer, and so cannot be accessed. In a specific implementation of HyperContext, the implementors of the HyperContext hypertext can use the most appropriate external representation for the profiles and interpretations according to the extent to which they wish to model the domain. For example, interpretations can be represented as vectors, a semantic network, a Baysian network, or a Petri-net, as long as inter-operability between HyperContext hypertexts with different external representations is supported.

Within HyperContext, interpretations are represented as vectors, so the interpretation of a text document in the context bottom is a vector of weighted labels. It is possible, although of limited use, for the interpretation of a document to be null, in which case all elements of the vector will be zero-weighted. Currently, documents of other media types, such as digital audio and video, are assumed to be represented as term vectors, so that it is possible to directly compare interpretations and queries across multimedia documents.

A collection of documents which, in the Structure Layer, are each interpreted only in the context bottom (and are therefore unlinked) can be queried through an information retrieval system. Although documents will be displayed to the user through the Presentation Layer if they are directly accessed, a user cannot browse through the hyperspace as there are no links between interpretations.

The process of linking an interpretation of one document to another includes the creation of a vector which describes the interpretation of the child document in a specific context. Typically, while the user is reading the interpretation of a document in context, she may know of, or may be informed of, another document which she considers to be of interest. She will select a region in the source document to act as the link source anchor, and will invoke the createLink function to select the document to be the link destination. If the destination document is not already represented in the Object Layer, the createProfile function is automatically invoked. Once the destination document's profile exists, an uninterpreted version of the document will be presented to the user, and she can choose the regions of the document which she considers relevant. createLink will call the createInterpretation function, which automatically generates a description of the destination document according to the relevant regions the user has selected. The user may also add labels of her own to, or remove labels from, the resulting interpretation, and change the relative importance of terms in the interpretation to reflect her subjective opinion. The next time the parent document is accessed in the same context by a user, he will see a link to that child document. However, should the parent document be accessed in a different interpretation, that link may not be visible.

The same document in different interpretations can be represented not only by different labels, but, when the same label appears in different interpretations, it can have different weights. This is a departure from normal vector-based representations of documents. Normally, a term weight reflects the relative importance of a term to a document, given the overall importance of the term in the entire document collection. The HyperContext framework does not specify how the term weights are derived, as this is the responsibility of an indexing system which is external to the framework. However, to give an example, the external information indexing system may use the *TFxIDF* method [79] to determine the relative importance of a weight in a document. TFxIDF is the product of the normalised term frequency (TF) of the term in the document and the inverse document frequency (IDF) of the term in the entire collection. So a term with a high frequency of occurrence in a particular document, but a low overall frequency in the document collection will be given a greater weight than another term of the same frequency which occurs frequently throughout the collection. Terms which are good at discriminating between documents are rewarded with higher weights than those terms which would result in a large number of documents being retrieved as relevant. Once a weight is allocated to a term in a document, that weight is always used to determine the extent to which a document is relevant to a query.

A different approach is taken in passage-level retrieval ([19], [93], [53]). Terms within the same document can have different weights depending on their relative importance to individual passages, such as sentences or paragraphs. The document collection is then ranked in order of relevance to a query according to the highest scoring passage in each document.

HyperContext employs region-level retrieval. When a user creates an interpretation, she specifies the regions in the child document which are relevant *to her*. She does not need to think about how other users might interpret the same document, she need only describe what she finds relevant about the document, given her own current needs and requirements. The resulting interpretation is associated with the link the user extended from the parent document she was reading. The document and link together form the new interpretation's context. Apart from this new link being available to future visitors to the parent node in the same context, the interpretation can be retrieved directly through information retrieval. HyperContext's region-level retrieval is similar to arbitrary passage retrieval, in that the same regions of a document in different contexts can have differently weighted terms. The contents of the selected regions are passed to an external indexing system to derive the initial label weights for an interpretation of a document. The label weights can then be modified to reflect the user's opinion. A user may modify the vector representing an existing interpretation of a document, by invoking modifyInterpretation, to add or remove labels or to change a label's relative importance in the interpretation.

The actual multimedia documents represented within HyperContext reside outside HyperContext, so users are normally unable to directly modify them. The effect of modifying or deleting a document referred to within HyperContext's Object Layer is unspecified.

Profile, link and interpretation management tools other than creation and modification are beyond the scope of this thesis. Also, although multimedia documents of a type other than text can implicitly form part of a HyperContext hypertext, they are not treated differently from textual documents. We assume that profiles in the Object Layer can represent non-textual multimedia documents, but that interpretations of such documents are represented by a vector of weighted terms. Chapter 10 discusses future work to provide additional profile, link, and interpretation management tools, and how multimedia documents might be treated to reduce the reliance on textual descriptions of them.

To summarise this section, before a document can be linked to, its profile must already be represented in the Object Layer. A document's profile is created using the createProfile function. A profile may be amended, usually only to add references to bind link source anchors to regions in the actual document, through modifyProfile. The createLink function extends a link from a source document in context to a specified child document. Simultaneously, createInterpretation creates a vector of weighted labels which represents the interpretation of the document in the given context. Interpretations may be modified through the modifyInterpretation function.

5.3 A HyperContext implementation model (1)

Chapter 7 describes in detail the prototype implementation of HyperContext used for demonstration and testing purposes. In this section, we give a brief example implementation.

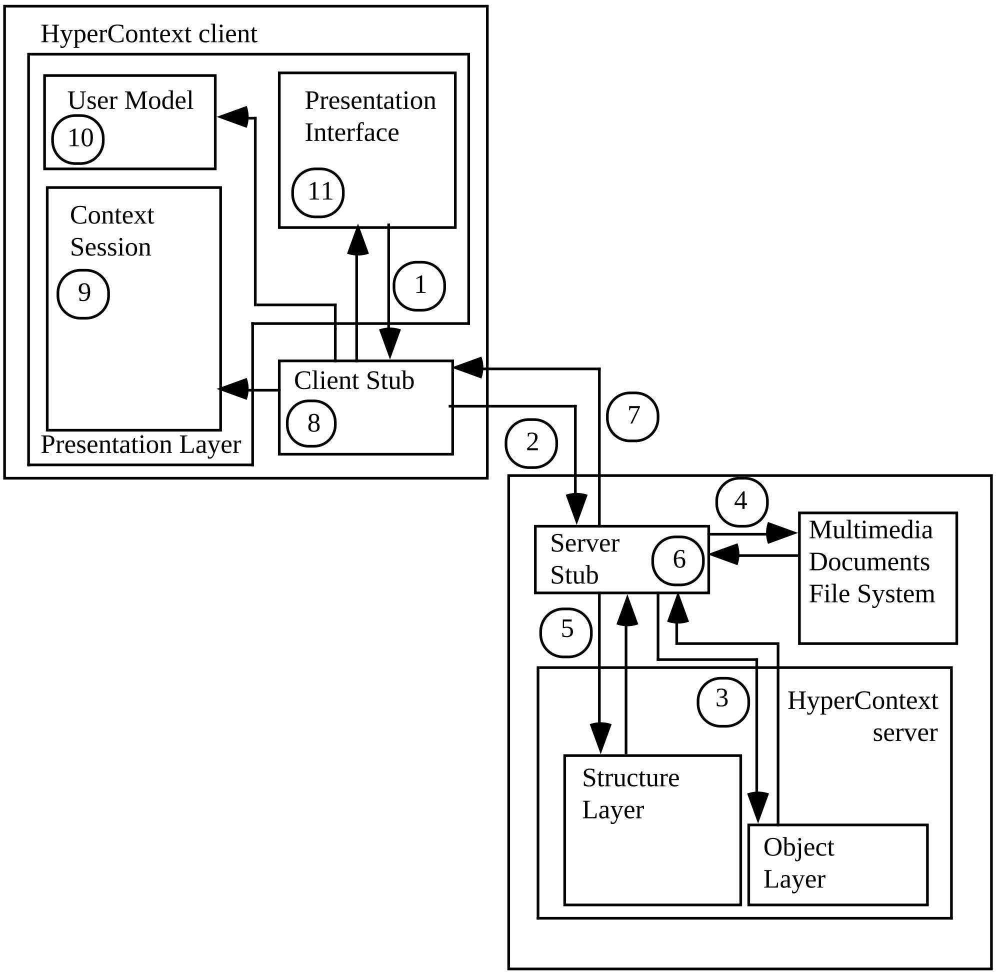

Figure 5.1: Partial implementation model of HyperContext

HyperContext may be implemented as a distributed system, with a distributed file system and distributed processing (figure 5.1). The HyperContext Object and Structure Layers contain document profiles and interpretations respectively which can be spread across distributed servers. A HyperContext server is responsible for maintaining resident document profiles and interpretations and delivering them to clients when requested. A HyperContext client provides the user with an interface to a distributed HyperContext hypertext, and maintains the short-term user model and context session which are part of the Presentation Layer. With reference to figure 5.1, when a user follows a link (1), the client generates requests for the appropriate document interpretation from the server (2). On receiving a request for a document interpretation, the server will retrieve the document profile to obtain the document's location and access method (3). It will then retrieve the interpretation (5) and the document itself (4) and applies the interpretation to the document, inserting link source anchors into the document where appropriate (6). The server will then return the interpreted document, and the document interpretation to the client (7). Apart from presenting the interpreted document to the user (11), the client extends the context session (9), and updates the short-term model of the user's interests (10).

To ensure inter-operability between HyperContexts with different underlying representations, it is necessary to provide a layer which is common to all implementations of HyperContext. This layer is the Presentation Layer, which in any implementation is part of the client. Individual HyperContext servers will receive generic requests from HyperContext clients. The servers can convert the generic requests into the specific formats required by the server-side processes. Information returned to the client must again be converted to the generic format. The generic representation of information in HyperContext is a vector of term weights for document interpretations and the short-term user model, and attribute-value pairs for document profiles.

5.4 Browsing through a HyperContext hyperspace

We now describe how a user can browse through HyperContext as an *adaptable* hypertext. A user can browse through a HyperContext hypertext by accessing an initial document in context. Following the example implementation details in Section 5.3, the user submits a request via the client for a node in context. The client will send a getDocument(N, C) request to the server hosting the requested node. The server will generate getInterpretation(N, C), getLinks(N, C) and getProfile(N) requests to obtain the document's description in this context, the out-links for the document, and link source anchor binding details respectively. The document itself may be retrieved using the method specified by the profile. The server will combine the out-links with the document and pass the modified document and the document's description in this context back to the client. If the specified context within which the user wants to view the document does not exist, then the server will send the interpretation of the document in the context bottom.

When the client receives the interpreted document and the document's interpretation, it will update the context session and perform any required further processing on the document prior to displaying it to the user. If an interpretation of the document in the specified context does not exist, the user is invited to create the interpretation, otherwise the document interpreted in the context bottom is display. Each time a user follows a link, the client-server cycle is executed.

The user is able to revisit interpretations using the context session as a history of the interpretations visited. The context session is modified accordingly.

Once a user is visiting a particular interpretation, she may request that the same document is displayed in a different context. In this case, the client will generate a getContexts(N) request of the server, which will in turn generate a contexts(N) request of the Structure Layer. The user will be presented with a list of all the document's contexts as node name (document title) and label name pairs. The user may also view the contents of any document which forms part of the context. To change the context of the current document, the user simply selects the new context from the list which results in a re-interpretation of the document. Unless the user instructs otherwise, this action is detected as a context switch (Chapter 4.5) and marks the start of a new context session.

At any point during a browsing session, a user can author links and interpretations of existing documents, and may even author a new document if she is running a HyperContext server.

5.5 Extending the metaphor

In the metaphor we have been using, a newcomer has arrived in town and we have compared the skills required of this person with those required by a hypertext user. We initially assumed that both the town and the hypertext were pre-built real structures but in Chapter 4.4 we introduced the possibility that the edifices are constructed at the same time that they are travelled through.

As a user wanders through a hypertext, or through the virtual town, there are a potentially infinite number of possibilities of what the user might see next. Ideally, we want the user to see something that is useful or interesting (as far as the user is concerned), although the serendipitous browser [22] may want to be surprised on occasion. In a hypertext or a town that is pre-built, the browser sees only what the designers have already decided should be seen. The only possibility to incorporate the browser's requirements is to blinker her (to prevent her seeing what she may not want to see) and to determine paths, from amongst those available, which may lead her to useful or interesting sites.

Consider that there are a number of interwoven worlds (partitions) in which a HyperContext hypertext or a virtual town can develop as the browser navigates. Every time a step is taken, all possible worlds which do not support the route taken thus far are definitely eliminated, and they can play no further part in the construction of the future route, rather like a game of chess in which the contestants have access to all possible configurations of the chess board. As movements are made on the board, there will be configurations, and therefore possible strategies, which are simply no longer available to each opponent. Past decisions affect future possibilities. This is also the case in a normal hypertext, although to a lesser extent. A browser may visit a document which has a number of out-links. There are now a number of possible paths which include this document. When the browser elects to follow one of the links, the paths which previously were a possibility, but now are not, are eliminated.

In HyperContext, as in the virtual town, links between documents, or streets between corners, exist because previous users or town-dwellers have decided that they are useful. They also describe each destination in context. Visitors to the virtual town and users of a HyperContext hypertext travel through their respective environments, confident that the links they are presented with have been relevant to previous users in the same context.

Both a HyperContext hypertext and the virtual town are potentially incomplete. A new user or traveller may require something different from the structure which has not been provided by other travellers before them. The virtual town and HyperContext need to provide a means of enabling further travellers to locate information which has not previously been required. In HyperContext, information is described in context. It is possible that when a user visits a particular interpretation, they require information which is not directly linked to the current interpretation, even though that information may exist somewhere in the hyperspace. Travellers and users need to be able to search for relevant information within their environment. Ideally, based on a user's past activities, their future needs can be automatically extrapolated and searched for on their behalf. The remainder of this chapter discusses different search methodologies as well as a method for automatically extrapolating a user's short-term interests based on her activities in the current context session.

5.6 Adapting the hyperspace to the user

So far, there are two main differences between HyperContext and a hypertext such as the WWW as far as a user is concerned. The first is that the links available from a document, and the destinations of those links, are dependent on the context in which the document is accessed. Secondly, a user is able to create links between arbitrary documents, even if she is not the owner of those documents. This section introduces adaptive features of HyperContext which actively assist the user in her search for information.

HyperContext provides users with individual support through adaptive information retrieval, automatic query formulation and reformulation, path and link recommendation services, and context-free "See Also" links ranked by relevance to a query extracted from the short-term user model.

In the HyperContext client, an interpretation *j* of a document *i* is internally represented by a vector $I_{j,i}$ of term weights $<w_0, w_1, ... w_N>$, where N is the size of the vocabulary. If w_i, where $0 <= i <= N$, is greater than 0, then term t_i associated with the weight w_i is a label in that interpretation. In vector-based statistical information retrieval systems, a document and a query can be compared to ascertain the degree of similarity between them. The greater the similarity, the more relevant the document is to the query. HyperContext utilises three approaches to information retrieval, each geared towards a particular search strategy. The first, Traditional Information Retrieval (TIR), is based on a classical centralised inverted index, which allows users to search for information without adaptive support. The second, Information Retrieval-in-Context (IRC), allows users to search for information which is directly or indirectly accessible from a document visited in the current context session, or from a document nominated by the user. As a result of performing an IRC search, the user can be guided to the relevant information while browsing. TIR and IRC are initiated by the user by submitting a query. The third search strategy, Adaptive Information Discovery (AID), softens the distinction between searching and browsing. While the user browses AID generates a query based on the model of the user's short-term interests to recommend paths to relevant information which can be reached from the user's context session. It also generates a context-free "See Also" list of links to recommended interpretations to enable the user to access other interpretations which may be relevant but which cannot be reached from the context session.

The short-term user model is also represented by a vector of term weights $<w_0, w_1, ..., w_N>$, where N is the size of the vocabulary. If $w_i > w_{threshold}$, then we say that the user is interested in the term associated with w_i, otherwise we say that the user is not interested in the term. The user model is normally automatically updated following a document access during a context session, but it can also be directly updated at any time by the user.

In HyperContext, information is organised so that children (interpretations) are always considered relevant to their contexts (see Chapter 6.5). An interpretation is represented by a vector of weighted labels. A user may want to know if she is geographically, or structurally, close to the information she seeks. A typical information retrieval system would return relevant documents without any indication of where those documents are in relation to the user's current position in the hyperspace. **Information Retrieval-in-Context** (IRC) takes a user query and a start location (a node in context) and searches for relevant information which can be accessed, directly or indirectly, from the start location. Relevance is a function not only of similarity between documents and the query, but also of distance (in terms of intervening nodes) between the location of the relevant documents and the start location. If the start location can be considered to be a landmark in our virtual town, then the user is asking for directions to an establishment from the landmark. This not only situates the information, but, because she is being directed to the information in terms of a location with which she is familiar, it reduces the opportunity to become disoriented. As an interpretation is described in terms that are definitely relevant to its context, and as an interpretation is created because a prior user has considered it reasonable to extend a link from one node to another, then if there is an interpretation, relevant to the user query, which can be accessed by following a path from the designated start node to the relevant interpretation, then we can assume that the interpretation is relevant not only because it is similar to the user query, but also because some rational entity has described it as relevant in that context.

Of course, it may be that an interpretation of information which is possibly relevant to the user either does not yet exist, or else it may exist in a different context. In the latter case, the interpretation may be similar to the query (using, for example, the cosine similarity measure), but there is no path from the user-specified start location to the apparently relevant information. As the hyperspace is constructed over time, there is no notion of a complete HyperContext hypertext. Parts of the hyperspace may be disconnected either because they are really irrelevant, or else because no prior user has had occasion to create a link between them. Yet, in this example, we have a case where a user has specified a query and selected a landmark from which to search. Assuming that the user is rational, then the user is insinuating that it is the case that the information she seeks is relevant, somehow, to the context of the landmark. Information Retrieval-in-Context will not locate information which is not directly or indirectly accessible from the nominated start location, because it is not sufficient for an interpretation to be similar to a query in order for it to be considered relevant in context. However, **Traditional Information Retrieval** (TIR) is a suitable search paradigm for this purpose.

TIR performs a context-free search for relevant information anywhere in hyperspace, without consideration for the context within which the information exists. Essentially, this means that a node interpreted in any context may be relevant to a query. It is quite possible that as well as finding several different documents relevant to the query, many interpretations of the same document may also be relevant. Although all relevant documents (and their interpretations) are ranked according to relevance, which relevant interpretation of a document should a user choose, given that selecting an interpretation of a document may well influence the parts of the hypertext which are subsequently accessible? This is currently an open question in HyperContext. In normal hypertexts, one of a user's bigger quandaries is perhaps "which link will lead me to the information I seek?". Whereas HyperContext in general may alleviate this problem, TIR has possibly created the new problem "which interpretation contains the link which will lead me to the information I seek?". There are several possible solutions. We can assume that in TIR a document is an end in itself, rather than the starting point for a browsing session. In this case, the user may be shown only the interpretation of the documents in the context bottom (although the ranking of documents may take into account the degree of similarity of the closest interpretation to the query). Once the user has accessed a document (interpreted in the context bottom), she is free to switch the context of the document to select the most appropriate interpretation (in which case, this will be the start of a new context session). Alternatively, the long-term user model may be employed to distinguish between interpretations, filtering out those which appear to contain information the user is usually not interested in. However, unless it is known in advance that the user's current task is related to a long-term interest, this approach has the disadvantage that the pursuit of a short-term interest can, indeed, contradict the same user's long-term interests. Finally, a more comprehensive solution might be to create *aggregate* nodes which summarise tracts of hyperspace in context. Now, not only would an interpretation need to be relevant, but it would also need to be represented in a relevant aggregate node. As we have already stated, this is a dilemma which is particular to Traditional Information Retrieval, and we are not committed to a particular approach. Some research opportunities for introducing typed nodes and links and aggregate nodes into the HyperContext framework are given in Chapter 10.3.2.

5.7 The TIR and IRC algorithms

5.7.1 Traditional Information Retrieval

TIR is likely to be used to identify a relevant document interpretation from which to begin a new context session, or to locate a document (which is already known to the user) so that the user can create a link to or from it.

TIR takes a user-specified query and compares it to all interpretations of all documents to identify those interpretations which are relevant to the query using the services of an external information retrieval (IR) system. The user is presented with a list of interpretations ranked in decreasing order of relevance.

Internally, a user query is represented as a vector of weighted query terms. HyperContext does not propose a minimum standard for the external services provided by the IR system, apart from the requirement that the results should be ranked.

If the IR system employed is a vector-based model of IR, then the user query will be compared to an inverted index of interpretations using a similarity measure such as the cosine similarity measure [78], and the list of interpretations presented to the user will be ranked in order of degree of similarity.

Hyperleaping to a document from the results of a TIR search is normally considered to be a context switch and is indicative of the start of a new context session. The user can, however, indicate to HyperContext that on this occasion it should be considered to be an extension to the current context session. The user explicitly tells HyperContext that the information she is accessing is relevant to her short-term interest. There are, therefore, very strong grounds for the user to create a link from an interpretation she has already accessed in her current context session to the document she has located through TIR search. If the user does not instruct HyperContext to extend the context session over the interpretation just accessed, then HyperContext will start a new context session, with the interpretation in context just accessed forming the root of the new session. The old context session is temporarily stored for further processing (to update the long-term user model, for instance, or for the user to re-visit later).
If the user has indicated that the context session should be extended over the interpretation just accessed, then HyperContext will recommend that the user extends a link from a document accessed in the context session to the relevant document (Section 5.2). The justification for this is that if the current user has a use for this association where previously one did not exist, then the chances are that a future user could benefit from the association too. The user is able to choose any interpretation in the context session to be the link source, although it is likely to be the interpretation accessed immediately prior to invoking the TIR search. The user will select a region in the source document to act as the link source anchor, and a new interpretation of the destination document is created by copying the selected interpretation of the destination into the new context formed by the source document and label acting as the link source. Additionally, the label weights in the interpretation corresponding to the terms specified in the user query are increased slightly, to distinguish it from the original interpretation. The user is then free to modify the new interpretation as she sees fit. Unless otherwise determined by the user, the out-links of the new interpretation are inherited from the original interpretation. Details of how and when the new interpretation will be available for future TIR searches is generally beyond the scope of HyperContext (as it is tightly bound to an external Information Retrieval system over which the framework has no control), but ideally it would be made available

immediately. The delayed availability of new interpretations is an issue only for TIR search. For the other search mechanisms and browsing, the new interpretations will always be available immediately.

5.7.2 Information Retrieval-in-Context

Like TIR, IRC also takes a user-specified query and compares it to interpretations of documents to identify those interpretations which are relevant to the query. Unlike TIR, however, IRC requires the interpretation to be reachable, directly or indirectly, from a document in context specified by the user. This may be the interpretation from which the user invoked IRC search; an interpretation the user has visited in the current context session; or any other specific interpretation. IRC is usually used when the user 'feels' that the information she seeks is geographically 'nearby', but its location is not obvious, and she wishes to be directed to it, if it exists.

The invocation of IRC reflects a subjective expression of what the user expects to find in the vicinity, even though she may not have visited this part of the hypertext previously. Once again, the chances are that if one user has this expectation, then others will also. If the information she seeks is located by IRC, she can be led to it by following a recommended path. She may alternatively decide to make the information more obviously visible, by creating a link to it from the document she identified as the start location for IRC.

Internally, an IRC query is a vector of weighted terms and a context-node pair which represents the start location. The query can be converted into the format required for an external information retrieval system to find all relevant interpretations of documents. HyperContext is responsible for processing the results of the search to ensure that they can be reached from the nominated start location (Chapter 6.6.2).

IRC can be implemented as a parallel breadth-first search through interpretations of documents in context, with the search root nominated by the user. The search can be controlled by a number of factors, including search depth, number of nodes checked, termination at first match or after a minimum number of matches, or a combination of factors. Once reachable relevant interpretations have been identified, the links leading to them can be recommended when a document is presented to the user from the interpretation nominated as the root. Chapters 7.6.3 and 7.4.2 contain examples from the HyperContext prototype of IRC search results and recommended links respectively.

5.8 The short-term user model

5.8.1 Deriving a model of the user's interests from an interpretation

So far, the framework describes a browsing session through a HyperContext hypertext, during which navigation support can be provided after a user-initiated Information Retrieval-in-Context search. Through Adaptive Information Discovery (AID), HyperContext is able to automatically determine a user's interests and can guide the user to relevant information. AID itself is discussed in Section 5.9. We first describe the short-term user model which provides AID with a representation of a user's short-term interests.

An interpretation is a vector of weighted labels which describes a document in a particular context. Each time the user accesses an interpretation, the context session, which is a list of context-node pairs, is updated to reflect the user action. The short-term user model is a synthesis of a user's interests which reflects the activity recorded by the context session. The short-term user model is also represented by a vector of weighted labels, where each weight is an estimation of the user's interest in the corresponding label.

The process of synthesising a user's interest is practically the reverse process of determining which documents are relevant to a query. In a vector-based model of IR, we plot the representation of the query into *n*-dimensional space. The query's nearest neighbours are the documents most similar to, and therefore most relevant to, the query. We know that two documents are related to some extent, not just because they are neighbours in *n*-dimensional space, but also because they are both related to the same query. If a user gives relevance feedback on documents which an IR system has established as being similar to the query, then the query can be automatically refined to generate a new query based on the original query and the vectors of the documents which were given relevance judgements. The Rocchio relevance feedback method [74] is commonly used in IR systems to re-formulate an initial user query by re-positioning that query in *n*-dimensional space according to relevance feedback provided by the user.

In HyperContext, an interpretation is known to be relevant to its context because a user has created a link between an interpretation of the parent and the child. It is important to remember, however, that a user creates an interpretation of a document not with reference to the interpretation's parent, but with reference to her own needs and requirements. When the user describes an interpretation, she needs to think of how the document is relevant to her needs, and not how it is specifically relevant to the document containing the link source anchor. The only information we have in a context (apart from the name of the context) is the description of some document which in the Structure Layer is linked to from some other interpretation of a document. The interpretation of the parent, however, is largely independent of the interpretation of the child, and *vice versa*. There is no guarantee that any labels in an interpretation will appear in its parent interpretation or those of its children. However, we do know that the interpretation of a parent and its children *are* related, because there is a link between them. In deriving a user model, we will derive a vector which represents the interpretations visited in the context session to summarise a user's short-term interest.

When we consider the relative importance of each term in an interpretation, we are able to determine the discrimination factor the term has across all interpretations of the same document. For example, consider that a document has five interpretations (indicating that the document exists in five separate contexts, including bottom). We will call these interpretations I_0, I_1, ..., I_4, with I_0 representing the interpretation of the document in the context bottom. Assume that there are five terms, t_0 to t_4, in the vocabulary, and assume that a binary weighting scheme is used to indicate that a term is relevant in an interpretation (value 1) or not relevant (value 0). Table 5.1 describes the vectors for each interpretation.

	t_0	t_1	t_2	t_3	t_4
I_0	1	0	1	0	1
I_1	0	1	1	0	1
I_2	1	0	0	1	1
I_3	0	1	1	0	0
I_4	1	0	1	0	1

Table 5.1**:** Weights of terms t_0 to t_4 in interpretations I_0 to I_4 of document I

In IR, the Inverse Document Frequency (IDF) measure is used to identify terms in documents with a high discriminatory power. If a term has a high Term Frequency (TF) in one document, then the temptation would be to say that the document is mainly about that particular term. So, if a term occurring most often in a document containing 100 words occurs 20 times, then it appears sensible to assume that the document is mainly about that term. If a user is searching for information about that term, then the document with a high frequency for that term appears to be highly relevant. However, in order to be able to rank documents in order of relevance, or similarity, to a query, we cannot make decisions about a document in isolation from the rest of the documents in the collection. In a vector for a given document, a term's weight reflects the relative importance of the term to the document (measured by Term Frequency), taking into account the overall distribution of the term in the entire collection (measured by the Inverse Document Frequency). This means that if a term has a high term frequency in a large number of documents, and only term frequency is used to reflect the relative importance of a term to a document, then many, perhaps too many, documents would be relevant to a query containing this term. On the other hand, a term which occurs infrequently in the collection, but a few times in one particular document would probably be discriminated against by the presence of the high frequency term, because its weight in the document vector would be low. Taking IDF into account means that a term with a high distribution across a collection of documents will only have a relatively high weight in those documents where it has a disproportionately high term frequency. On the contrary, a term with a low document frequency will be rewarded with a high weight whenever it occurs in a document. TFxIDF effectively normalises term weights.

In table 5.1, the term weights are a reflection of the relative importance of a term in an interpretation. We know, from earlier discussions, that the relative importance of a term in an interpretation can be influenced by the user creating the interpretation, or by other users modifying the interpretation. However, for the time being, we will assume that the weight of a term is derived by a function such as TFxIDF, where the number of documents is not the same as the number of interpretations (and IDF is dependent on the number of documents, not the number of interpretations of a document). TF, however, reflects term frequency in an interpretation (so that a term in different interpretations of the same document can have different weights). Furthermore, the TFxIDF result is thresholded, so that a value of or above the threshold yields a weight of 1, and a value below the threshold is given a weight of 0.

Each interpretation that a user accesses during a context session effects the model of the user's short-term interests so that a query can automatically be generated on the user's behalf to locate and guide the user to relevant information. We will always assume that the interpretation that the user is currently visiting is of interest to the user but it does not contain (all of) the information that she seeks, unless she informs us otherwise. What does the interpretation she is currently visiting tell us about her short-term interest? When a user creates an interpretation, she describes the document using terms which make the document relevant *to her*. When a user accesses an interpretation we assume that the terms used to describe the interpretation are also relevant to her. However, unless the interpretation contains the information she seeks we know that she will continue browsing, so the user interest in the current interpretation cannot merely be a copy of the interpretation's vector. The path that the user has followed has led to a particular interpretation of a document. That document may have other interpretations. Can any information be deduced from the differences between the interpretations? Can the distribution of terms in the interpretations of a document provide information about what distinguishes the interpretation selected by the user from the others? There are several approaches available, some term dependent and others term independent. A term independent approach would consider the appropriateness of each term independently of all others, whereas a term dependent approach would consider groups of terms at a time. If we consider that the interpretation accessed by the user is I_1 (from table 5.1), and we are attempting to generate a user model *u* based on the interpretations of I, then a term independent approach might consider the weight of t_0 in *u* as a function of the weights of t_0 in the interpretations of I. On the other hand, in a term dependent approach the weight of t_0 in *u* could be a function of the pairs of significant weights $<t_0, t_1>$, $<t_0, t_2>$, $<t_0, t_3>$, $<t_0, t_4>$ in each interpretation of I. A term independent approach is adopted in HyperContext although term dependent approaches are also well-suited to statistical methods of information retrieval and user modelling [59].

In statistical models of IR, a query is usually re-formulated using the Rocchio relevance feedback weighting formula given in formula 5.1

$$Q' = Q + \alpha \frac{\mathbf{1}}{|R|} \sum_{D_j \in R} D_j - \beta \frac{\mathbf{1}}{|NR|} \sum_{D_j \in NR} D_j \qquad \text{Formula 5.1}$$

where α and β are constants which determine the overall importance that will be given to terms in relevant and non-relevant document sets respectively, *R* is the set of user-selected relevant documents, *NR* is the set of user-selected non-relevant documents, D_j is a document vector, and finally, *Q* is the original query. If a document collection could be automatically precisely partitioned into the sets of relevant and non-relevant documents, then the Rocchio method would generate an optimal query for ranking all relevant documents above all non-relevant documents.

Can the Rocchio method, or an adaptation of it, be used in HyperContext to generate a user model based initially on the selected interpretation of a document, over all interpretations of the same document, and subsequently, on the combination of interpretations in the context session? A document can have tens, to hundreds or thousands of interpretations (some of which may be identical, but existing in different contexts). As in IR, we do not want to automatically modify the query so that it would return all and only those documents that the user has already selected as relevant. We want the new query to return previously unseen documents which are hopefully more relevant to the user than the ones he has already seen.

To use the Rocchio method we need to identify sets of relevant and non-relevant documents. The user, however, accessed a single interpretation by following a link to it. He is probably not even aware of the overall number of interpretations that exist for that document, let alone want to give relevance judgements for them. Indeed, as long as the user is presented with the right document, he is probably not overly concerned about which particular interpretation yielded the document. To use the Rocchio method effectively, we also need to have an original query to modify, as well as determine appropriate values for α and β.

Although a user may not be too concerned with the actual description of the document she views, the interpretation is essential for HyperContext because it is a way of discriminating between possible descriptions of that same document, and because the choice of interpretation effects the way that the hyperspace will be partitioned. We know that the document is not precisely (perhaps not even imprecisely!) the document she seeks, otherwise the context session would end at this document. Given that in the Rocchio method an original query is modified to generate a new query, perhaps we can use the description of the document in the current context as the initial query in the Rocchio method, with a view to modifying the description to more accurately reflect the user's interests. In formula 5.1, Q would be the vector representing the interpretation accessed by the user.

The next decision to be made concerns the sets of relevant and non-relevant interpretations of the document which will be used to modify Q. We know that the user's past actions have led to a particular interpretation of a document rather than any other interpretation of the same document. We know, then, that the set of relevant interpretations contains at least the interpretation the user is at. As yet, we cannot make any relevance judgements about the interpretations the user has not accessed, nor can we expect her to make a judgement. It seems redundant to use the accessed interpretation as the only relevant interpretation (especially if the interpretation is used as the original query). It also seems heavy-handed to consign all unchosen interpretations of the same document to the non-relevant set.

All interpretations of a document refer to the same document. We somehow need to identify those interpretations that the user would consider relevant and those that would be considered non-relevant, without any conclusive evidence such as relevance judgements made by the user. The interpretation accessed by the user has a position in n-dimensional space in relation to the other interpretations of the same document. We could rank all interpretations of the document according to similarity to the accessed interpretation (using the cosine similarity measure), and place the most similar $|R|$

interpretations into the relevant set and the least similar |*NR*| interpretations into the non-relevant set. We can then apply the Rocchio method so that labels in the original query which are common to the relevant interpretations and missing from non-relevant ones would have their weights reinforced, while labels in the original query common to non-relevant interpretations would have their weights decreased. Labels common to both sets would be more or less unaffected.

Let us consider using the approach just described in a normal statistical model of IR. Automatic query modification based on documents which are most and least similar to the original query would have the following effect. The user submits a query to the IR system containing, for example, 100 documents. The IR system, using the cosine similarity measure, ranks the documents in order of similarity to the query. Without user feedback, the IR system then uses the Rocchio method to automatically modify the user's query. The 20 most similar documents are placed in the set of relevant documents, and the 20 least similar documents are placed in the set of non-relevant documents. After Rocchio is applied, the query will become more similar to the top 20 ranked documents, and less like the bottom 20 ranked documents. Although the gap in similarity between the query and the top and bottom ranked documents will increase, it is unlikely to result in a change of ranking order. However, let us consider that the 100 documents in the collection represent a training set for a much larger collection. It is now feasible that the new query applied to the larger collection would result in not only a different ranking order from that which would have been obtained by the original query, but also in a different set of relevant documents. Similarly, the new query in HyperContext will not be used to rank or select from amongst the interpretations from which the new query is derived, but it will be used to select relevant interpretations of other documents.

The next approach we consider, although we have already alluded to it being redundant and heavy-handed, is to place the selected interpretation into the set of relevant interpretations, *R*, and all interpretations of the documents will be put into the set of non-relevant documents, *NR*. To reduce redundancy, the selected interpretation is not used as the original query, *Q*. Instead, *Q* will be a vector of zero-weighted terms, and so can be omitted from the formula. α will be 1. *NR* contains all interpretations of the document, rather than all but the selected interpretation. The main reason for this is the term weights are averaged in the Rocchio method, so as |*NR*| increases the influence of a single interpretation decreases. Using all interpretations means that the average weights of each term in the vector can be pre-calculated and stored for efficiency. β will be the *degree of similarity* of the selected interpretation to the average interpretation in *NR*. The value of β is justified because we want to synthesise the essential differences between the selected interpretation and the interpretations that were not selected. If the selected interpretation approximates the average interpretation then the terms in the selected interpretation are less able to act as discriminators, but as the selected interpretation becomes more distinct from the average the terms become good discriminators.

The modified Rocchio method to synthesise a user's interests based on the selected interpretation is given in formula 5.2. We are not generating a modified query, so rather than producing Q' , we produce $I_{salient}$, the *salient interpretation* representing the synthesised user interest.

$$I_{salient} = \alpha I_{sel} - \beta I_{ave} \qquad \text{Formula 5.2}$$

In formula 5.2, $I_{salient}$ is the synthesised user interest in the current document, α is normally 1, and I_{sel} is the selected interpretation. β is the degree of similarity between I_{sel} and I_{ave} measured by the cosine similarity measure. I_{ave} is given in formula 5.3,

$$I_{ave} = \frac{\mathbf{1}}{|I|} \sum_{I_j \in I} I_j \qquad \text{Formula 5.3}$$

where I is the set of all interpretations for the document, and I_j is an interpretation in I.

In Chapter 6.7.1 we reject this specific method for estimating a user's interests based on the distinguishing characteristics of the accessed interpretation from other interpretations of the same document. One of the reasons is that if the accessed interpretation was completely relevant to the user's requirements, and satisfied her information need, then the context session would probably come to an end upon accessing this interpretation. Given that the context session is extended, we can deduce that at best the accessed interpretation was only partially relevant to the user. If we were to predominantly use the distinguishing characteristics of the accessed interpretation on which to base our estimation of their interest, then the user model could very quickly represent the information in which the user is not that interested. Instead, when we synthesise the user's interest we treat the accessed interpretation as a description of the document that is not as appropriate as some other interpretation of the same document.

The description of the interpretation that we will include into the model of the user's short-term interests is based more on the features which are common to all interpretations of the document, and less on features which are particular to individual interpretations. We still derive the average interpretation I_{ave} and the accessed interpretation I_{sel}, but now the salient interpretation will contain those features of the average interpretation which are not present in the accessed interpretation (formula 5.4). The terms in the accessed interpretation are weighted according to how similar they are to the terms in the average interpretation, so that if the accessed interpretation is identical to the average interpretation, then we cannot extract any useful information from the average interpretation.

$$I_{salient} = \alpha I_{ave} - \beta I_{sel} \qquad \text{Formula 5.4}$$

Chapter 6.7.1 justifies this approach, given results observed during a study comparing these two (and other) approaches (Chapter 9.4.5).

5.8.2 Deriving a model of the user's short-term interests from the context session

We have so far derived a method of synthesising a user's interest from a single interpretation using formula 5.4. However, this single interpretation is only one of many that the user has accessed during the current context session. A salient interpretation is derived for each interpretation that the user accesses during the context session. We now need to relate the salient interpretations to each other to derive a user model from which we can automatically generate a query to locate information of interest to the user and to lead her to that information.

We assume that as the context session grows, we can learn more about the user's interest. The early stages of the context session probably cannot tell us as much about the user's interests as can the later stages, unless the user makes her interest explicit. A context session can be initiated in one of two ways: either by user specification of a document in context (by providing details of the document and a context), or else by specifying a user query in TIR or IRC search. We first consider the case where a user has accessed an interpretation by specifying a document and context.

When a user initiates a context session by specifying a document and a context within which to interpret the document, we assume that the document is a landmark from which the user will begin browsing. If this is not the case, that is, the interpretation is the one which the user seeks, no harm is done because the context session will end at that point. At the start of a context session, we assume that the landmark is distantly related to the information that the user ultimately seeks. As the user browses, accessing one interpretation after another by following links, we assume that she gradually edges closer (in descriptive terms) to the information she seeks. As the HyperContext hypertext may be incomplete we cannot assume that as a user browses she actually approaches the information she seeks, because that information may not be accessible from the context session. However, each interpretation she accesses adds some information to what we can assume about the user's interests. The amount of influence that the resulting salient interpretation will have on the overall user model representing the user's short-term interests will be on a scale of confidence: salient interpretations at the beginning of a context session will have less overall influence than salient interpretations that occur towards the end of a session. The scale is user-adaptable, but is generally 0.125, 0.25, 0.3, 0.4, 0.6, 0.7, 0.8, for the first, second, third, to seventh, salient interpretation of the context session. If the context session is more than seven long, then all salient interpretations from the seventh onwards are weighted with a confidence factor of 0.8. The weighted salient interpretations in the context session are then averaged and this average is used as the short-term user model (formula 5.5).

$$u = \frac{1}{|CS|} \sum_{i,j=1}^{i=7,\ j=|CS|} scale_i \bullet I_{salient,j}$$ Formula 5.5

In formula 5.5, u is the user model, $|CS|$ is the length of the context session, $scale_i$ is the ith weight in the scale of confidence, and $I_{salient,j}$ is the jth salient interpretation in the context session.

If the user initiates the context session by means of a TIR or IRC search, then HyperContext has a significant amount of information about the user's interests, because the user will have made it explicit in the form of a query. The original query can be incorporated into the salient interpretation generated for the first interpretation accessed by the user following a search (formula 5.6). As the first interpretation is related to the original query, $I_{salient}$ will be derived from the pertinent features of I_{sel}, compared against the average interpretation of the document, I_{ave}. If the user continues browsing from the first accessed interpretation by following links, the salient interpretations of subsequent documents in the context session will be derived using formula 5.4.

$$I_{salient} = \gamma Q + \alpha I_{sel} - \beta I_{ave}$$ Formula 5.6

In formula 5.6, γ reduces the influence of the original query in the salient interpretation. In TIR and IRC search, following a query, the user is presented with a ranked list of relevant interpretations. The formula is applied when the user accesses one of the interpretations in the list. When the user initiates a context session by hyper-leaping to an interpretation, we assume that the interpretation is only remotely indicative of the user's interests. When the context session is initiated after a TIR or IRC search, then we can have greater confidence that the selected interpretation is more indicative of the user's interests. Therefore, although we will still use a scale of confidence in progressive salient interpretations, the first salient interpretation of the context session can be weighted with a higher confidence factor. We also assume that the number of steps required to locate relevant information starting from a relevant interpretation is less than the number required starting from a landmark, so the confidence scale is 0.5, 0.6, 0.7, 0.8, with subsequent salient interpretations weighted at 0.8 confidence.

5.8.3 Abstracting a query from the user model

In vector-based models of IR, the vector representing the user model would simply be used as the query. Interpretations would be compared to the query and the most similar would be returned as relevant. In other models of IR, the user model would need to be converted into the required format. A Boolean model of IR may take the highest and lowest weighted terms from the user model, and AND and NOT the terms respectively. If the query is expanded using a thesaurus, then terms can be ORed if they are synonyms.

5.8.4 Openness of the short-term user model

The short-term user model is user inspectable and modifiable. The user model consists of weighted terms. The number of terms is likely to be huge, and most of the terms are likely to be zero-weighted. Consequently, only significant terms are shown to the user, and usually only a subset of these.

The user model can be modified either by changing the relative positions of terms in the list; by increasing or decreasing term weights; by 'deleting' terms (which effectively zero-weights them); or by 'adding' terms (which effectively promotes terms which were not shown to the user because they were not significant enough).

5.9 Adaptive Information Discovery

Adaptive Information Discovery (AID) acts upon a query extracted from the model of the user's short-term interests to advise the user of relevant information. If there is a path to the relevant information from the current context session, then the user can be recommended links to follow, otherwise relevant information not reachable from within the context session can be dynamically linked to as "See Also" references. AID can use IRC to locate relevant information which can be reached from the current context session, and TIR to provide a list of "See Also" links (after removing references to interpretations already located by IRC).

The user is not constrained to following a recommended link. If the user follows a link which has not been recommended, the user model is re-evaluated and a new query is extracted. The user may then be recommended a link and a list of "See Also" links to information relevant to the new query.

The user can invoke TIR and IRC searches while AID is active at any time during a context session. The search terms provided by the user are re-weighted in the user model to reflect the importance given them by the user. The search is run twice - once using the user supplied query and once using a query extracted from the modified user query. The results are merged (using data fusion techniques, for example, [91]), with the effect that interpretations which have a high relevance in both searches are promoted above those which do not. In the merged list, interpretations which are more relevant to the user-supplied query than the extrapolated query have a higher ranking.

AID can be activated and deactivated by the user, and any search parameters (such as search depth) that AID requires can be provided by the user. AID does not normally make "See Also" and link recommendations during the early stages of a context session if it was initiated by a hyperleap to an interpretation, although the recommendations are immediate if the context session is initiated following a TIR or IRC search. The AID recommendation delay is user adaptable.

AID recommends paths to relevant information when previous users have described information that is relevant to the current user, and it is reachable from the node in context that the user is currently visiting. AID recommends context-free "See Also" links to interpretations which HyperContext determines may be relevant to the user, based on the user model, but which do not have the added confidence of being reachable from the user's current location in hyperspace.

5.10 Summary

We have described how users can adapt a HyperContext hyperspace and how HyperContext provides users with adaptive navigation support.

Users can adapt the hyperspace by adding new documents, and creating and modifying document profiles, links, and document interpretations. Users create interpretations of documents by describing how the document is relevant to their needs and requirements. A context is created for the new interpretation by extending a link to the interpretation from some other document interpretation. Interpretations exist in a HyperContext hyperspace because previous users had considered it both relevant and useful to create them.

Adaptive navigation support is provided by HyperContext through two new search paradigms and link recommendation. Information Retrieval-in-Context (IRC) permits a user to automate a search for information that she conceptually considers to be close to her current location in the hyperspace. IRC restricts its search to interpretations of documents which can be accessed from a document in context. Adaptive Information Discovery (AID) automatically generates a query on a user's behalf and searches for relevant information. Traditional Information Retrieval (TIR) is a non-adaptive search paradigm based on classical information retrieval techniques which can be invoked directly by the user to search for relevant information throughout the hyperspace. The user can be guided to relevant information found by IRC and AID if it is reachable from the user's context session. AID also uses TIR to search for potentially relevant interpretations which are not reachable from the context session and recommends them as dynamic "See Also" links.

A model of the user's short-term interests is constructed during a context session by deriving and combining salient interpretations of each document interpretation that a user accesses. A scale of confidence is used to weight each salient interpretation according to its position in the context session prior to its inclusion in the user model.

The distinction between browsing and authoring, and browsing and searching is reduced by providing assistance with link and interpretation creation as users browse and by Adaptive Information Discovery respectively.