

---

# Implementing an Open Link Service for the World-Wide Web

---

L. A. Carr<sup>1</sup>, D. De Roure, W. Hall & G. Hill  
Multimedia Research Group,  
University of Southampton, SO17 1BJ, UK

---

<sup>1</sup> Email address: lac@ecs.soton.ac.uk

## ABSTRACT

*Links are the key element for changing a text into a hypertext, and yet the WWW provides limited linking facilities. Modelled on Open Hypermedia research the Distributed Link Service provides an independent system of link services for the World-Wide Web and allows authors to create configurable navigation pathways for collections of WWW resources. This is achieved by adding links to documents as they are delivered from a WWW server, and by allowing the users to choose the sets of links that they will see according to their interests. This paper describes the development of the link service, the facilities that it adds for users of the WWW and its specific use in an Electronic Libraries project.*

Key words: Links, hypertext, open hypermedia, link services, authoring, HTTP proxies.

## 1. INTRODUCTION TO LINK SERVICES

Hypertext environments have always provided more functionality than word-processors, and the hyper- part of the environment has been in the form of links. Links are the connections between different parts of text and are often seen as buttons (a graphic or highlighted word on which the user can click) which cause a new piece of text to be displayed. Links come in many forms: they can embody the concrete connection between two adjacent pages, sections or footnotes in a document or they can represent the organizational connection between a paragraph in a document and an annotation by a reviewer. They can even express the abstract connection between a concept mentioned in one document and an explanation of that concept in a glossary, dictionary or encyclopaedia. Links are informational entities in their own right, and can be stored, processed and used independently of the documents to which they refer.

Open hypertext systems [Malcolm *et al.* 91; Davis *et al* 92] are those hypertext environments that allow links to be freely used in this way, connecting spreadsheets with videos, interactive online sessions with news services or literary archives with terminological databases. They aim to act as an underlying hypertext *link service* (a term first used by Pearl [89]), rather than provide a closed environment in which to present information. Such a link service aims to allow hypertext facilities to be accessed from any available application, thus acting as a service component of the user's environment. In order to provide such a facility, link information must be managed separately from documents, so that links may be applied to documents in any format.

The authors' Microcosm hypertext environment [Davis et al 94] provides these facilities for a user's information environment, but limits them to documents that are locally accessible. By contrast, the World Wide Web has global scope, and is undoubtedly the most successful hypertext system ever, but its provision of linking facilities has not been achieved using an open mechanism as described above. Instead the HTML document format is relied upon to allow the author to express a document's connections as a part of the production of the document. Although links may be created to documents other than those in HTML and image formats, such links are usually dead ends, and there may be no way to follow any further links *e.g.* links from spreadsheet documents. There is also no way for additional links to be made available by third parties, as all link information is embedded in documents.

The Dexter model [Halasz and Schwartz 1990] is a way of describing hypertext systems according to their component services, in particular the mechanisms that they provide for handling documents and links. An extended version of this model [Grønbæk and Trigg 1996] has been used to contrast document-embedded WWW links (known as *location specifiers*) and the links of an open hypertext link service (*reference specifiers*). By adding a *reference specifier* linking service to the normal WWW document transport server it is possible to employ an open hypertext approach to the authoring and management of World Wide Web hypertext documents.

The development of open hypertext and hypermedia systems has highlighted a number of advantages over closed systems that embed link information into documents. The most significant examples are briefly described below.

## 1.1 Reduced Authoring and Maintenance Effort

The most obvious advantage is the ability to improve authoring efficiency, and subsequently to aid in the maintenance of hypertext documents. By using the different linking strategies found in Microcosm [Davis *et al.* 92], for example generic links and information retrieval facilities, it is possible to rapidly create a useful set of links.

In particular, the use of generic links allows common links to be authored only once — wherever the source selection of the link occurs, the link is available, including any documents subsequently made available. Typically such links would be created on names of people and places, or common terms, to provide access to more detailed information. In a closed system, such links need to be created wherever the source term appears in a document, and new documents also need to be linked into the system manually.

Storing links in a link service reduces maintenance requirements as it allows changes made once to a central database to be immediately effective wherever the link is available. This can reduce problems frequently encountered in the WWW, such as link fossilization and decay [Hill *et al.* 1995]. Link editing can be more efficient as it is more convenient for a user to search for all links that satisfy a particular criterion, rather than to look for them on a document-by-document basis. Also, automatic processes can be set up more efficiently, for example to check whether the destination of any of the links have been deleted.

Although care must be taken to avoid the risk of inappropriate linking on synonyms, the generic form of linking described above also lowers maintenance effort by reducing the number of links stored in the system. A single generic link that can be attached to many places where the right key phrase appears is an abstraction over a set of links that attach themselves to a specific location in a specific document. A truly generic link that anchors itself to a given concept independently of the language used to express that concept would be even more useful!

## 1.2 Enhanced Reader Experience

Another aspect of a link service is the integrated interface offered. Rather than provide hypertext facilities as a single application, a link service aims to provide underlying hypertext facilities to the user's whole environment. Thus hypertext navigation may be used as a general method for managing and traversing the user's information space.

In addition, the type of linking described in the previous section allows the user a more flexible approach to link traversal. Rather than rely on having every link highlighted by the system, the user is also able to select arbitrary items and query the system for possible links—thus creating a 'reader-led' navigation paradigm [Hall 1994].

Readers may also be provided with the facilities necessary to create their own links, allowing them to annotate material which in other systems they would not be able to annotate and freeing them from a hypertext structure created purely by designated authors. If these databases may be shared with other users, collaborative authoring of hypertext resources is enhanced.

## 1.3 Alternative Views

A powerful mechanism made possible through the use of separate link management is the provision of multiple link databases for the user to select from. Thus the user may select link sets that reflect the context in which he wishes to investigate a particular set of documents. For

example, in a university department with a particular set of resources, students might select from tutorial-oriented linksets, whilst departmental staff might use an entirely different linkset to support research activities. This ability to impose different ‘webs’ of links over a given collection of documents was first demonstrated by the Intermedia hypertext system [Haan *et al.* 1992], where it was demonstrated to particular effect in educational applications.

Another possibility is a separation between information provider and link provider. At present, hypertext material is usually delivered with links inextricably bound to the associated material. A link service can help to overcome this restriction, by providing the facility to apply completely different link sets to a set of documents, or conversely to apply existing links to new documents not available when the links were originally created. This makes it possible for third parties to offer pure linking services which end users may apply to any documents which they can access, breaking the common binding between content and link structure.

Finally, this facility can also aid in more efficient management of hypertextual information. If a variety of ‘webs’ are to be applied to a particular set of documents, changes to the document set are easier to make if the link information is managed separately. If link information had to be embedded in the documents, then many different document sets would have to be maintained in order to provide the alternative link structures. Similarly, if new documents are introduced, existing link information need not be embedded in them to facilitate navigation, as links are immediately available.

## 2. A SIMPLE INTERACTIVE LINK SERVICE

It is clear from the discussion above that the WWW, as it is used at present, is unable to provide many of the desirable features of a hypertext link service. However, the open nature of the framework upon which the WWW is based means that it is possible to extend the hypertext model of the WWW to support the more advanced features required of a link service. In effect the WWW infrastructure, now widespread and readily available, can be used as the communication framework for a distributed hypertext link service.

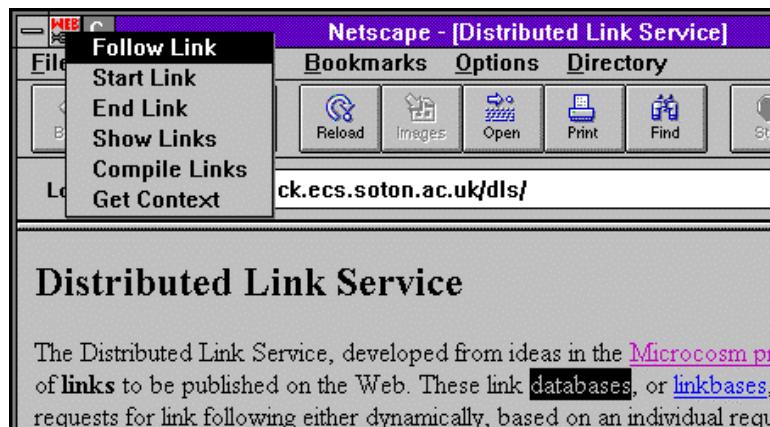
We have developed the Distributed Link Service (DLS) as such a system. It is able to work in conjunction with existing WWW resources to support an additional underlying link service, which is able to provide the features described in the previous section. This system is based upon our experiences developing the Microcosm hypertext system [Davis *et al.* 1994]. Like Microcosm, the DLS utilizes a variety of link database processes to offer flexible hypertext functionality to a wide range of end-user applications.

The DLS [Carr *et al.* 1995] is composed of two parts: the server facilities that are accessed via the WWW, and the client interface that works in conjunction with a WWW browser.

### 2.1 Link Server

The link server facilities of the DLS that were implemented first as CGI scripts invoked by a standard WWW server are now implemented as modules of a pseudo-WWW server. This pseudo-server interacts with clients as if it were a normal WWW server, using enough of the hypertext transport protocol to allow normal interaction with a browser, but it does not store or return any documents. Instead, modules are available to allow the creation, traversal and editing of links, which are stored in a number of link databases. The databases use an SGML style mark-

up, and record the source and destination attributes of the link, the type of the link, its creation time and a link description.



**Figure 1a: A user requests a link from the link service using the client interface**

There are several different link database categories supported by the system, at the most general level are server databases, which apply whenever the system is queried. Link databases may also be provided for a group of documents, or a particular document. In addition, a variety of ‘context’ link databases are available which the user may select from. By choosing a different context, the user may adjust the available link set to best suit their current information requirements. The user is also provided with a personal link database in which they may create private links that only they have access to.

The server receives details from the DLS client of the user’s selection, the document in which the selection was made, and the context selected. The *followlink* module determines which link databases are required, and gathers these together to satisfy the request. Like Microcosm, the system supports the use of generic links, which allows links to be applicable beyond the scope in which they were originally created.

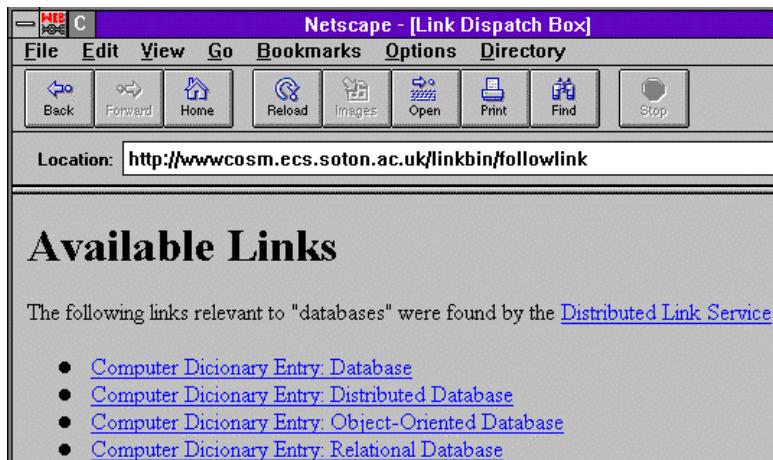
The *editlink* module provides an HTML form which allows the user to select from the available link databases and edit the individual links via an HTML form, perhaps changing the default link description or updating the link’s destination. The *createlink* module accepts details of start and end points for a link, and enters a new link into a specified context link database or into the user’s personal link database. The *context* module provides a list of the different context link databases available on the server and can be used by the client to present a menu of contexts to the user.

All of these modules can be driven by an explicit command coded as a URL and sent to the link server as a normal HTTP request. Alternatively the server can return a form to the browser, and the user can fill in the fields to support a particular action. However, form-filling is not a convenient user interface for actions like creating a link or editing a large link database because there are too many pieces of information which have to be entered manually (source and destination URLs, link title *etc*).

## 2.2 Client Interface

To provide a more direct interface than form filling for many of the link service operations, a DLS client was produced for the PC, Mac and UNIX platforms. The client presents itself as a set of menus attached to the title-bar of any document viewer. It is a simple utility that formulates

DLS requests and communicates these to the selected link server via a WWW browser. The client allows the user to select a predefined context, or topic of interest, from one menu and reacts to DLS requests from a second menu (figure 1a). Details of the selection the user has made, the document in which this selection is found, and any selected context is encapsulated as an HTTP request and communicated to the WWW client browser using the platform's appropriate IPC facilities. The link server returns the results of the DLS request to the WWW browser, which then presents the results to the user. For example the result of a Follow Link request might offer a list of appropriate links (figure 1b), or indicate that no links were found.



**Figure 1b: The link server responds with a page of available destinations**

### 3. AN INTERFACELESS PROXY LINK SERVICE

A client of the link server can extract details from any application (not just a Web browser) and create link requests that are passed onto the link server via a Web browser. The DLS can therefore provide links for data maintained by applications that may not have their own hypertext linking mechanisms. Alternatively, it is possible to combine the chosen links in with the original document, if the document's format is capable of representing hypertext links. Hence an option for producing hypertext material is to develop it using the interactive clients described above, and then to *compile* a chosen set of link databases into a specific set of document resources (currently in HTML, RTF or PDF format) which will then be independent of the link service. By varying the compilation parameters, different webs of links may be produced over similar material for different audiences.

To provide a convenient interface to the facilities of the DLS, the interactive client required the ability to find information such as the current selection and the identity of the current document from many different applications. Eliciting such information required a 'non-standard' software solution at a time when WWW browsers and inter-application communication were constantly evolving. Further complications came from revisions in operating system implementations, and meant that it was difficult to make the client work consistently as the WWW environment was constantly upgraded.

Independently of software development problems, feedback from users of the DLS (publishers and information providers) expressed the opinion that the link service should not change the user's experience of the WWW by making them have to use extra menus or learn any new

concepts. It was also seen as desirable that no software should be downloaded to the user's computer, and no special installation procedure should need to be supported.

Hence an alternative, 'interfaceless' approach was investigated: to make the link service transparent to its users by embedding it in the Web's document transport system, compiling links into documents *as they are delivered to the browser* by a specially adapted WWW proxy server.

This approach requires no extra client software for the user, which is an immediate practical benefit, but it does suffer from a number of disadvantages. Firstly, abandoning the client it is impossible to create a link by the usual method of making a selection and choosing *Start Link* from a menu. It also changes (perhaps for the worse) the browsing paradigm from "reader-directed inquiry" to "click on a predefined choice" [Hall 1994]. Secondly, this behind-the-scenes link compilation is applicable only to documents which are delivered via the WWW and which are coded in well-understood document formats that can themselves support some form of hypertext link. These requirements abandon some of the advantages of the open system previously described, since there are relatively few document formats into which links can be embedded.

The 'clientless' server is therefore a compromise, giving up some of its open hypertext philosophy in order to gain simplicity of delivery. Note that the link service is still open in the sense that links can be applied freely to documents, it is just the universe of document formats which has been closed.

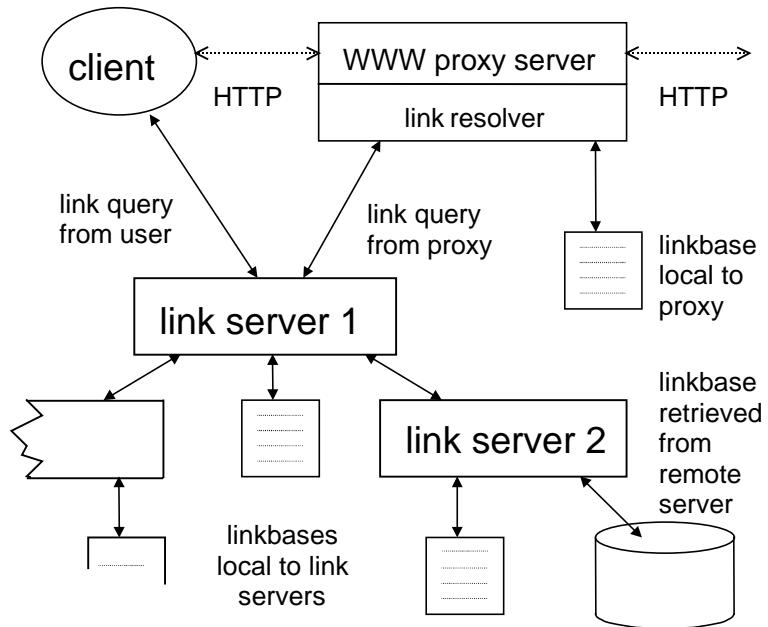


Figure 2: Distributed Link Service network model

### 3.1 Link Service Architecture

The rest of this section describes our general model for link resolution on a network. The diagram in figure 2 depicts a user at the client accessing documents via the adapted WWW proxy server described above, with the proxy accessing a local linkbase in order to resolve links. It also shows the user making queries interactively (perhaps following links, or authoring new ones) and these are serviced by link server 1, which also has a local linkbase.

Local linkbases are the simplest approach to link resolution, but the link resolution agents embodied in the proxy and the link server have two other options:

1. They can pass the query on to another link server. In figure 2, the proxy server can pass queries to link server 1, and link server 1 can pass queries to link server 2. This is useful both for concurrency (obtaining links from multiple servers simultaneously) and for redundancy (working with multiple servers with the same linkbases,).
2. Instead of sending the query towards the linkbase, they can bring the linkbase towards the query. This is achieved by retrieving the linkbase data from an appropriate server, as shown for link server 2. Linkbase data is just another document type and it can be cached using the same techniques as any other document, which promotes scalability.

Finally, the link resolution agent itself may be mobile. Instead of data moving to the agent, the agent can then move to the data. This model is a topic of current research but has not been realized in any practical DLS implementations at this time.

Note that a simple client might talk to the first process in the diagram, but more sophisticated clients could incorporate the functionality of any of the processes described here—these are “heavyweight clients”.

### 3.2 Link Control Panel

Since the interfaceless link server requires at least some initial configuration (for example choosing applicable sets of link databases), a method for communicating with the server has been developed. This takes the form of a kind of a “link remote controller” which is an HTML form displayed as a separate window and whose results are interpreted by a module in the link server (figure 3).



Figure 3: Link Service “Remote Controller”

The controller establishes a session (a binding of a user and host together with a set of link server parameters) which is used to control the behavior of the link server from that point in time onwards for that particular user. It is intended that the user will invoke the controller just once to set a preferred configuration, and only again afterwards to adjust the configuration—links will always be added automatically to the documents according to the last settings of the controller.

## 4. CONTROLLING LINKS

The purpose of the controller is to give to the user the ability to choose how links are displayed and used within the processed documents. Previous versions of the software elevated links to first-class objects by giving them a status independent of their containing documents, but the DLS now allows the user to directly manipulate links to control presentation and navigation. This section describes these facilities.

### 4.1 Link Inclusion

The control panel in figure 3 gives the user the ability to choose which of the server's installed linkbases are to be combined with requested documents, or to completely bypass the link compilation if a 'normal' document viewing mode is required.

The control panel provides a greater degree of control over the linking process, enabling the user to specify in some detail which link databases are switched on and off as the user browses in and out of a number of document resources. By doing this the user can control the kinds of linkbase that are used at such a point (e.g. internal navigation through a resource *vs.* citation of documents external to the resource).

The Open Journal Framework [Carr *et al.* 1996] makes use of this kind of control panel to help the user navigate through large suites of collected but separate Internet resources, all integrated by the use of linkbases. By introducing a model of Internet resources (collections of documents and associated link databases) and aggregations of these resources (collections of collections of documents and associated link databases), it is possible to define the user's location in a document space, and hence to know what hypertext actions are applicable at each point in that document space. If the user travels outside all known resources (e.g. to a colleague's personal home page), then the option still remains to apply the most general links or else to have the link server refrain from applying any links.

Without this model the same sets of link databases are applied to any document which the user sees.

### 4.2 Link Presentation

Once a link is selected for inclusion in a document by virtue of its presence in a chosen linkbase and its applicability to the current document (often determined by a simple keyword matching operation) the DLS inserts the link according to a specific presentation format (as seen in figure 4).

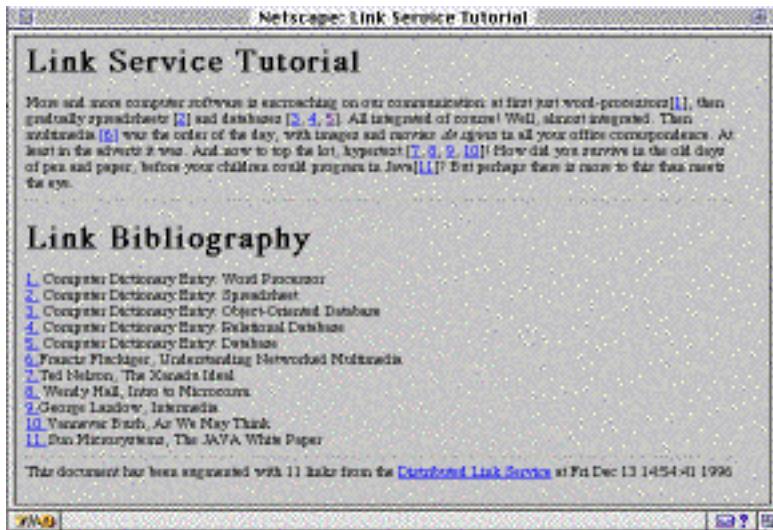


Figure 4: DLS adding bibliographic style links to a Web page

The recent standard for Cascading Style Sheets for HTML documents [Lie and Bos 1997] allows the presentation of many document features to be controlled by visual parameters such as font, size and color. WWW links in HTML documents are normally tightly bound to previously marked-up anchors, and so a style-sheet's only option for parameterizing link presentation is to change the typographic attributes of the (fixed) anchor. By contrast, the DLS has complete freedom to choose how to elaborate a link by binding it to any suitable site in the document. The DLS may apply the link to any part of the document's contents, or may invent a new piece of content to act as an anchor in the form of a distinguishing marker or a more general annotation.

DSSSL, a related standard for document styles and semantics [Angerstein 1996], and its proposed WWW equivalent XSL [Adler *et al.* 1997], operate on a model in which documents are processed in two phases: firstly to rewrite and re-order their components and secondly to apply formatting operations to the revised components. This model allows new content to be created for a document as it is processed and is the kind of model which the DLS employs, in contrast to Cascading Style Sheets.

Using the controller, links can be formatted according to the following styles given that a fragment of the document's content has been chosen as a link site

default	the matched content is chosen as a direct site for linking. The chosen text is displayed as all normal WWW links (determined by the browser options, but often underlined in blue for an HTML document, or given a box outline if a PDF document).
footnote	add a footnote marker (asterisk or dagger) after the identified link site. The marker and not the text carries the web link.
citations	add an annotation which looks like a bibliographic citation after the identified link site. The marker and not the text carries the web link, but not to the ultimate destination, rather to the matching entry in a 'link bibliography' appended to the foot of the document. The entry in this pseudo-bibliography contains a link to the ultimate destination.

indirect	the link is presented in any of the above styles, but does not go directly to the ultimate destination. Instead, it invokes the link service to produce a page describing the ultimate destination (or destinations if there is more than one link which matches a given position in the document). The indirect option is particularly useful if there are likely to be a large number of links on any given key word, as it stops many pieces of content being invented to site each of the necessary direct links.
----------	---

The link formatting styles are global attributes: all links will be presented in whichever style is chosen, although it might be useful to have the facility where particular links (or collections of links) are formatted in a particular style. Although this decision is currently left to the tastes of the user (or system administrator) in fact the most important role of presentation is not decoration but discrimination. A user must be easily able to distinguish between links which are added by the server and those which are native to the document (*i.e.* links calculated by a computer *vs.* links inserted by the author). In addition the user would like to be able to judge the likely pertinence of each calculated link without having to follow it. For this reason the DLS can further modify the way in which links are emitted by varying their color or font style. This enables brighter links correspond to more direct and specifically authored links created by a human author, whereas duller colors correspond to general links created by a simple dictionary lookup or a statistical lexical operation.

### 4.3 Link Prioritization

To allow the user to discriminate between different links the server must have some concept of the pertinence of different kinds of links. This measurement can be used by the server in other ways: to determine how it is to cull links from a potentially over-annotated document.

At any one time there may be many dozens of link databases active, providing links of various levels of relevance. An important task of the server is to throttle over-zealous link producers and allow the user to choose the overall proportion of link items to document content, the maximum number of links to appear on each key phrase or the maximum number of links to a particular resource.

### 4.4 Link Access

Thus far, the link control facilities have been concerned with rendition: what to put in a document and how. However, if links are first-class objects we can legitimately turn our attention not only to their provision but also to their usage.

The accustomed user interface for links (click and go) is convenient in some respects: uncomplicated and immediate it allows the user directly to jump between information resources. However, it is in other ways a very unnatural activity when compared with the sequence of actions it mimics in the physical world away from the can-do atmosphere of cyberspace.

When readers attend to a journal article in a library, they do not immediately follow each citation and cross-reference that is encountered as the text is digested. Instead, they may make a mental note to follow it up at a more convenient time, even scribbling it down on a pad. Only when the paper has been read to the readers' satisfaction will the attention then be turned to the cited material.

In other words, users have prior experience with a reading model in which they evaluate the content before they evaluate the links, whereas the WWW and most other hypertext environments provide an environment in which the user is repeatedly interrupted, stacking up unfinished document contexts to be returned to later. Although the hypertext environment itself does not force the user to switch contexts to the linked material, the lack of support for any other browsing protocol often makes it the line of least resistance.

In computing terms, the hypertext browser imposes a stack-based document evaluation modality onto the user, replacing a natural queue-based information processing methodology. This stack-based approach is impractical in the real world because of the significant time taken to change document contexts when compared to the Web.

As well as providing mechanisms for controlling the prioritizing and presentation of links, the DLS supplies a mechanism to help control the link following process, making it more like the real-world experience described above. It does this by providing an auxiliary "navigation planner window" adjacent to the users' browser window such that users can drag link anchors from the browser window onto the planner window, where they are displayed as icons. The icons can be moved around the window, clustered together according to the user's own informal classification scheme and subsequently double-clicked to make the browser display the relevant Web document.

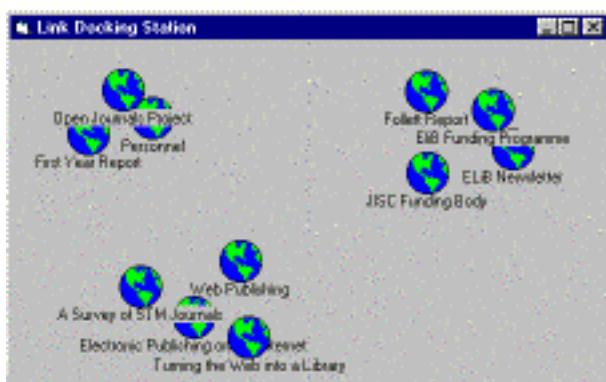


Figure 5: DLS link access auxiliary window

As such an electronic notepad has been created for jotting down interesting places to visit (a speculative bookmark list). However, a secondary function of the notepad is to pre-fetch the referenced documents while the user finishes browsing the main document, so that the reader really does get instantaneous access when the follow-up texts are examined. In fact, the referenced URL and all embedded data should be fetched, so that documents containing frames or images can display without delay.

A further function of the navigation notepad is to help the user to put the process of link following into perspective by explicitly recording the context in which the current document and its linked items are being read. This can be particularly helpful since reading is done not in an intellectual vacuum, but as part of a process of writing, of note taking and of goals and strategies for creating other documents.

## 5. USING THE LINK SERVICE

A significant experiment in the use of the Distributed Link Service has been the Open Journals project [Carr *et al* 1996; Hitchcock *et al.* 1997], an electronic library experiment which uses various kinds of external links to integrate independently published academic journal resources. An 'Open Journal' is a collection of documents (journal articles, online topic-specific databases, useful educational resources and bibliographic services) together with collections of associated links that together bind the separate resources into an integrated super-publication.

Several subject-specific open journals have been produced, including one in Biology which is based around the familiar keyword-style generic links (figure 6) and one in Cognitive Science, based around citation links (figure 7). In contrast to generic links, which are explicit connections between two documents triggered by matching keywords, citation links are the implicit connections between a citation, the bibliography item and the cited article's metadata stored in a bibliographic database. Citation links are triggered not by a simple keyword matching action, but by a more complicated database lookup based on the ability of a new module of the link service to parse the citation data held in the body of the document.



Figure 6: Biology Open Journal using Keyword Links

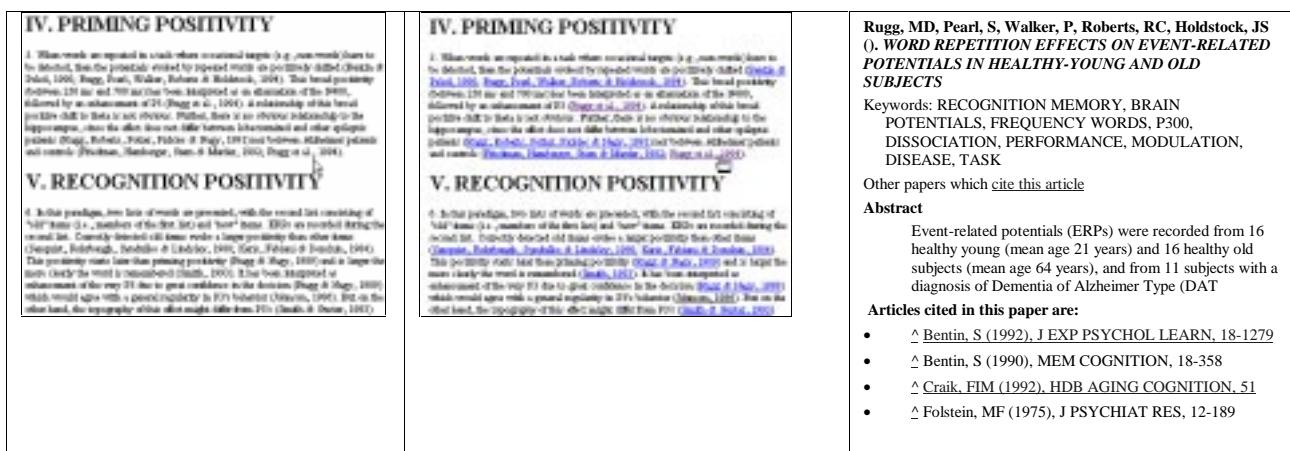


Figure 7: Cognitive Science Open Journal using Citation Links

Initial feedback from the project's focus groups confirms that the link service is easy to set up and use. The majority of users prefer to have an indirect style of presentation for keyword links (section 4.2) because it provides them with a list of destinations that they can evaluate before they make a decision to follow up any particular link. The most frequent requests are for links to a greater range of resources than the dozen or so publications that form the core of the project. (Links to research databases are seen as particularly important to biologists, and links to a more complete set of metadata is the key issue for cognitive scientists.) Tempering this desire for more links is the requirement to be able to more accurately control the kind of links which are displayed to the user—some keywords were felt to be too broad or elementary to be useful.

## 6. RELATED WORK

The concept of external link services has been a familiar part of the hypertext research community for many years, especially in the area of Open Hypermedia Systems. Although the WWW is a closed system according to OHS classifications, the provision of a link service provides a degree of openness, along with the Web's own use of external document viewers. According to the flag taxonomy of open hypermedia systems [Østerbye and Wiil 1996] the DLS is an incomplete session manager, taking on the responsibility of link availability *i.e.* controlling which links should appear as link markers. Link activation, the other normal responsibility of a session manager, is usually handled by the WWW browser itself, although it is possible to make the DLS perform this task by selecting 'indirect' links from the control panel in figure 3.

The use of an Open Hypermedia Protocol (OHP) for such an environment has been recently discussed [Davis *et al.* 1997] but is not directly comparable with the situation on the WWW. In this case, the native clients of the WWW speak a standard protocol to a WWW server not to receive linking information, but to receive a document that only indirectly contains links. The DLS server masquerades as a WWW server in order to resolve an explicit request for a document and an implicit request for links and translate the results back into data that the client is expecting, *i.e.* a document. This can be seen as a variant on the standard OHP architecture where the link server has the additional responsibilities of procuring the document and merging the links into the document; in the OHP scenario these two tasks are accomplished by the client. Although the two situations are not exactly congruent, it is possible that the OHP protocol could be effectively used between the components of the distributed server described in section 3.1.

The Hyperwave system [Maurer 1996] implements a link service but requires documents to be imported into it from the WWW to gain the benefit of its added links. It does however provide advantages for link editing over an Open Hypermedia environment because all link data can be kept strictly synchronized with the changing document corpus. This is difficult to arrange in an open system because the link service is largely independent of the document management facilities.

Based on the extended Dexter model of hypertext described in the introduction, [Grønbæk *et al.* 1997] describes a very similar link service mechanism to the DLS in which the links are maintained by a separate server, but combined with the text document by a Java applet embedded in the user's browser. This corresponds to the so-called 'heavyweight client' model in which part of the server processing is devolved to the client.

Research in the WWW community [Brooks *et al.* 1995] has been focussed on the use of *transducers* that intercept the flow of communication between a client and server, modifying the request or response in some way. Such transducers have been used to experiment with adding

extra functionality to the document server in the form of annotations, indexes and change marks [Meeks *et al.* 1996]. The DLS server acts very similarly, modifying the WWW server response (the document) by adding extra data to it, but for reasons of efficiency it does so as a “mutant proxy server” rather than an extra processing node on the communications stream. The DLS also breaks the transducer model by allowing the client to communicate directly with it by using the “Link Remote Controller” described in section 3.2.

Others have investigated the use of independent meta-information servers but to provide collaborative annotations for WWW pages rather than links [Röscheisen *et al.* 1994]. This is actually quite similar in concept to the DLS, as links can be easily provided within an annotation framework (in fact the DLS currently provides support for annotations through the inclusion of extra data in the link databases).

Early work on spatial metaphors for organizing and classifying hypertext material [Marshall *et al.* 1991] inspired a prototype of the link access facilities described in section 4.3 by one of the authors [Carr 1995]. Further work in the hypertext community on the use of spatial hypertext [Marshall and Shipman 1993; Marshall *et al.* 1994] has resulted in a commercial system (Bernstein’s Web Squirrel [1996]) which provides improved link access facilities for the WWW.

The WAIBA project of the OSF [Brooks *et al.* 1996] also produced software tools for improving link access for users of the Web. In particular, a Table of Contents agent produced a structural overview of a Web hierarchy to help the user make decisions about how to browse that part of the information space.

## 7. CONCLUSIONS

A hypermedia link service can provide important functionality for any information system by increasing the navigation options for users without increasing the problems of information maintenance. In conjunction with the Web it provides a powerful tool with which to address the restrictions of embedded links that are fixed with the publication of a document.

We have shown that a simple link service can be implemented using standard Web browsers and servers and described a revised implementation that works without the need for any additional client software. We have also shown how a link service can provide novel user-centered facilities for link presentation, discrimination and navigation and have demonstrated a use of the DLS for providing post-hoc integration to a set of academic journals.

We are continuing to develop the Distributed Link Service following our open hypermedia philosophy, adopting new browser and server technologies as they become available. Future work includes trialling various of the architectures described to permit a link service to scale to multiple hosts, an investigation of client-side link resolution (the ‘heavyweight client’), link caching on proxies, and multicasting to multiple link servers. Experiments are also being performed on effective ways of controlling the presentation of links.

One of the most important practical requirements of a link service is the ability for non-technical users to easily create useful collections of links. Thus far links have been created individually or as a group from existing meta-data and keyword extraction techniques. The former kinds of links are highly pertinent but very expensive to produce whereas the latter are cheap but sometimes too general to be practically useful. We are currently testing some simple tools that allow authors to

quickly describe and catalogue WWW resources in a way that allows an automated process to produce sets of links to those resources.

A significant practical shortcoming of the DLS is the requirement to retrieve and apply links to the document while it is being delivered to the user. Any delay in the processing of the links therefore means that the user has to wait for the document. Although this has not proved to be a significant problem for explicit databases of links, the more complicated processing required for citation linking requires certain trade-offs to achieve the response times that WWW users expect. Furthermore, any kind of on-the-fly linking in the semantic domain (links between concepts) rather than in the purely lexical domain (links on text features) would slow the delivery of the document beyond the limits of acceptability. Therefore a new agent-style interface is being designed for the link service, so that collections of links can be offered to the user as they are discovered by various independent link processing modules all acting asynchronously. It is hoped that this will allow the WWW to deal with the inherently abstract connections between texts mentioned in the introduction, rather than the point-to-point 'jumps' which it has allowed thus far.

## ACKNOWLEDGEMENTS

The work on PDF linking was carried out by the Electronic Publishing Research group at Nottingham University as part of the Open Journals project, JISC grant ELP2/35 from the Electronic Libraries program.

## BIBLIOGRAPHY

Adler, S., A. Berglund, J. Clark, I. Cseri, P. Gross, J. Marsh, G. Nicol, J. Paoli, D. Schach, H.S. Thompson and C. Wilson (1997), "A Proposal for Extensible Style Language (XSL)," Note, World Wide Web Consortium, MIT, 545 Technology Square, Cambridge, MA <URL: <http://www.w3.org/TR/NOTE-XSL.html>>.

Angerstein, P. (1996), Summary of the Document Style Semantics and Specification Language (DSSSL), International Standard 10179, International Standards Organization Document ISO/IEC JTC1/SC18/WG8 N1427.

Bernstein M. (1996) Eastgate Web Squirrel FAQ <URL: <http://www.eastgate.com/squirrel/FAQ.html>>

Brooks, C., M. Mazer, S. Meeks and J. Miller (1995), "Application-Specific Proxy Servers as HTTP Stream Transducers", The Web Journal 1, 1, 539-548, O'Reilly and Associates.

Brooks, C. (1996), "Wide Area Information Browsing Assistance Final Technical Report", Technical Report, The Open Group Research Institute. <URL: <http://www.osf.org/www/waiba/papers/y2report/y2report.htm>>

Carr, L. (1995), "Structure in Text and Hypertext", Ph.D. Thesis, University of Southampton, UK. <URL: <http://journals.ecs.soton.ac.uk/lacethesis/>>

Carr, L., D. De Roure, W. Hall and G. Hill (1995), "The Distributed Link Service: A Tool for Publishers, Authors and Readers", The Web Journal 1, 1, 647-656, O'Reilly and Associates.

Carr, L., D. De Roure, W. Hall and G. Hill (1996), "Open Linking Services", Computer Networks and ISDN Systems, 28, 7/11, 1027-1036, Elsevier

Davis, H., W. Hall, I. Heath, G. Hill and R. Wilkins (1992), "Towards an Integrated Information Environment with Open Hypermedia Systems," In ECHT '92, Proceedings of the Fourth ACM Conference on Hypertext, Milan, Italy, November 30-December 4, 1992, ACM Press, pp. 181-190.

Davis, H., S. Knight, W. Hall (1994), "Light Hypermedia Link Services: A Study of Third Party Application Integration," In Proceedings of the Sixth ACM Conference on Hypertext, Edinburgh, Scotland, September 1994, ACM Press, pp. 41-50.

Davis, H., A. Lewis, A. Rizk (1996), OHP: A Draft Proposal for an Open Hypermedia Protocol, presented at ACM Hypertext '96 Conference, Open Hypermedia Systems Workshop, <URL: <http://diana.ecs.soton.ac.uk/~hcd/protweb.htm>>.

De Roure, D., L. Carr, W. Hall and G. Hill (1996), "A Distributed Hypermedia Link Service" In Proceedings of the Third International Workshop on Services in Distributed and Networked Environments (SDNE96), IEEE Computer Society Press 1996.

De Roure, D., W. Hall, H. Davis and J. Dale (1996), "Agents for Distributed Multimedia Information Management," In Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM'96), London, UK, April 1996, pp. 91-102.

Grønbæk, K., N.O. Bouvin and L. Sloth (1997) "Designing Dexter-based hypermedia services for the World Wide Web," In Proceedings of the Eighth ACM Conference on Hypertext, Southampton, UK, pp.146-156.

Grønbæk, K. and R. Trigg (1996), "Toward a Dexter-based reference model for open hypermedia: Unifying embedded references and link object," In Proceedings of the Seventh ACM Conference on Hypertext, Washington, USA. pp. 149-160.

Haan, B. J., P. Kahn, V.A. Riley, J.H. Coombs and N.K. Meyrowitz (1992), "IRIS Hypermedia Services," Communications of the ACM, 35, 1, 36-51.

Halasz F. and M. Schwartz (1990), "The Dexter Hypertext Reference Model", In *Proceedings of the Hypertext Standardization Workshop Jan 16-18 1990*, National Institute of Science and Technology (Special Publication 500-178), pp. 95-133.

Hall, W. (1994), "Ending the Tyranny of the Link", IEEE Multimedia 1,1 pp. 60-68.

Hill, G.R. Wilkins and W. Hall (1993), "Open and Reconfigurable Hypermedia Systems: A Filter Based Model," Hypermedia, 5, 2, pp. 103-118.

Hill, G., W. Hall, D. De Roure and L. Carr (1995), "Applying Open Hypertext Principles to the WWW", In Proceedings of the International Workshop on Hypermedia Design, Montpellier, France, 1-2 June 1995.

Hitchcock, S., L. Carr, S. Harris, J. Hey and W. Hall (1997) "Citation linking: improving access to online journals," In *Proceedings of Second ACM conference on Digital Libraries*, Philadelphia, pp.115-122.

Lie, H.W. and B. Bos (1997), *Cascading Style Sheets Designing for the Web*, Addison-Wesley.

Malcolm, K.C., S.E. Poltrack, and D. Schuler (1991), "Industrial Strength Hypermedia: Requirements for a Large Engineering Enterprise," In Hypertext 91: Proceedings of Third ACM Conference on Hypertext, San Antonio, TX. ACM Press, pp.13-24.

Marshall, C.C., F.G. Halasz, R.A. Rogers and W.C. Janssen (1991), Aquanet: a hypertext tool to hold your knowledge in place. In Proceedings of Hypertext '91, San Antonio, Texas December 16-18, pp. 261-275.

Marshall, C.C. and F. M. Shipman III (1993), "Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext," In Proceedings of Hypertext '93, Seattle, Washington, November 14-18, 1993, pp. 217-230.

Marshall, C.C., F.M. Shipman and J.H. Coombs (1994), "VIKI: Spatial Hypertext Supporting Emergent Structure". In Proceedings of the ACM European Conference on Hypermedia Technologies, Edinburgh, Scotland, Sept. 18-23, pp.13-23.

Maurer, H., Ed. (1996), Hyper-G now Hyperwave: the next generation Web solution, Addison-Wesley.

Meeks, W., C. Brooks and M. Mazer (1996), "Transducers and Associates: Circumventing the Limitations of the World Wide Web," in Proceedings of the Conference on Emerging Technologies and Applications in Communications 96, Portland, Oregon.

K. Østerbye and U. Wiil (1996), "The Flag Taxonomy of Open Hypermedia Systems," in Proceedings of the Seventh ACM Conference on Hypertext, pp. 129-139.

Pearl, A.(1989) "Sun's Link Service: A Protocol for Open Linking," In Hypertext '89 Proceedings, Pittsburgh PA, 1989, pp. 137 – 146.

Röscheisen, M., C. Mogensen and T. Winograd (1994), "Shared Web Annotations as a Platform for Third-party Value-Added Information Providers: Architecture, Protocols and Usage Examples" Technical Report CSDTR/DLTR, Computer Science Department, Stanford University, Stanford, CA 94305, USA. <URL: <http://www-diglib.stanford.edu/rmr/TR/TR.html>>