Academic Information Management: an Open Linking Approach

Gareth Hughes, Wendy Hall, Tony Hey and Les Carr
Department of Electronics and Computer Science, Southampton University
Highfield, Southampton, United Kingdom. SO17 1BJ.
{gvh, wh, ajgh, lac}@ecs.soton.ac.uk.

Abstract: This paper describes a Web based document management system developed as a Lotus Domino application and the continuing research work of providing users with a variety of link services and agents that enhance the basic content of the system. The system is designed for use by administration personnel in an academic environment taking into account the wide variety of systems and methods already in use. Users do not need to know how to author Web pages as the source material for the system are files produced by common word processors. The system features a number of management tools to complement this concept written into the Domino application. The document management system is complemented by the use of an open linking service to dynamically cross-reference the documents.

Introduction

It is becoming commonplace for academic departments to maintain a Web site of official information for staff and students. The creation and maintenance of such a site is far from trivial. The AIMS (Academic Information Management System) project sets out to make it a very simple matter for the people who create paper documents within a department or organisation to contribute them directly to an automatically generated Web site. Advanced linking technology is used to enhance these online documents with as much relevant information as possible. The AIMS server becomes a fully featured Web-based repository and archive for the information. The user is simply required to provide a minimal amount of information when submitting a word-processed file from a Web form. AIMS has been designed around the concept of making use of the content of files rather than just being a Web-based file store. Users are expected to continue to work as normal on their own machines and submit finished versions of documents to AIMS.

An external link service enhances AIMS Web pages by embedding links on words or phrases found in AIMS pages. The links are not embedded in the documents on the server but are added to the Web pages on the fly by the link service. For example, a set of links can be automatically created that link staff names and user id's to user's home pages. The link service and other agents attempt to reflect the structure of the department complementing the information contained in the documents.

The AIMS Document Management System

This section contains a summary of the key features of the document management aspects of the system as implemented using a Lotus Domino server. Documents are submitted as actual files using a simple Web form. A small amount of information is required about the document but every effort has been made to make it a quick and easy process to upload a file. A complex LotusScript program manages the creation of a document in the Notes database and conversion of the actual content of the file into the internal Notes format whilst also storing the original file. Numerous formats are supported including Microsoft Word and WordPerfect. This ability to work with the content of a variety of file formats is the main reason why Lotus Domino was chosen as the application development environment for the project. From this database of rich text information the Domino server can generate Web pages on the fly. The overall design is targeted towards making a simple and fast process for puttting normal word-processed documents online with the minimum of effort.

Features of the document management system include :

- Documents are accessed from various generated listings designed for the application.

- Version control is supported. Users can update documents in the system as required. The system maintains the relationships between versions of documents.
- A custom designed search interface allows searching of the document content.
- AIMS stores all original files as well as storing the Notes version of the content.
- Automatic generation of Adobe Acrobat PDF files for all word-processed documents. This allows users to obtain print accurate copies of documents.
- Online editing of document details, future versions of AIMS may allow direct editing of the document content using a Java applet.
- Special facilities for large documents. These are stored in sections with a special table of contents document defining the relationship between them. This concept has been extended to allow users to define collections of documents such as 'All the minutes for 1997'. This is implemented using dynamic database lookups so the collections are continuously updated. The code library for this could be used in other Lotus Notes applications.

AIMS are now in use within the author's department and being evaluated by a number of others at Southampton and other UK Universities. Currently there are three Domino servers running 5 separate AIMS databases.

The Application of Open Linking

The consequence of the AIMS design is that documents do not contain embedded links in the way that a normal Web site would. The initial research goal of the project was to provide useful links on this content in a way that is easy to maintain. This is achieved by making use of an open hypermedia link service developed within the group. Over time the service has been implemented in two flavours though both work along similar principles. The first is the Distributed Links Service (DLS) [Carr et al 1995] [Carr et al 1998] [Carr et al 1998a] [Hill et al 1995] developed within the group mainly funded by the Open Journals Project [Hitchcock et al 1997] [Hitchcock et al 1998]. The commercial version, Webcosm, is developed by Multicosm Ltd, a company originally founded to exploit the Microcosm [Fountain et al 1990] system. Each brings differing advantages for the AIMS project. Webcosm has a larger feature count and will be used when the linking service is made available to users of AIMS whereas the DLS source is controlled by the group and the system can be altered at a low level for experimental purposes.
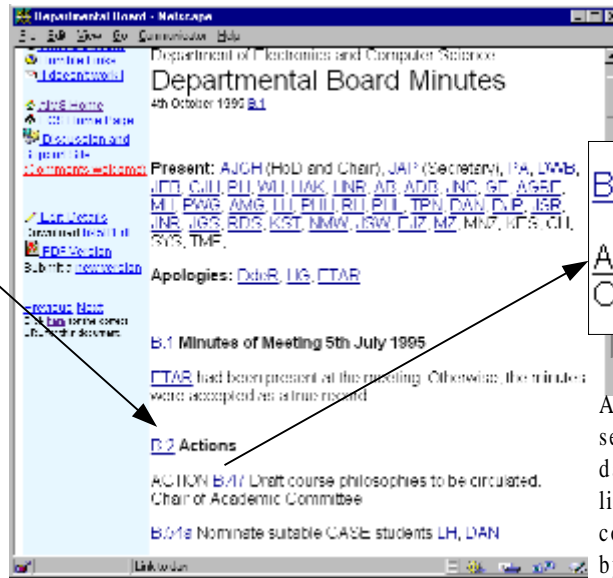
The link service is implemented as a Web proxy. The user configures their browser to use the link service as their Web proxy. When the browser requests a page it will ask the link service for the page. The link service will retrieve the page from the server (AIMS) and annotate the HTML with extra links before passing the enhanced Web page back to the user's browser. Links are stored on the link server in link databases or linkbases. An entry in a linkbase consists of the word or phrase to make an anchor, the destination URL and a description of the link. When the link service is in use any occurrence of the word or phrase in pages the user views will become a link to the destination document even though this link is not in the original Web page. This is a generic or glossary link as first implemented in the Microcosm hypermedia system. An example of an entry in a linkbase would be to link the acronym of a research group in the department to the home page of the research group. A link of type local restricts the anchor to only occur in a single document.

The main linkbases in use with AIMS are a set of links for all undergraduate courses to the relevant course page, a large linkbase linking user id's and user names to home pages and a linkbase with a variety of acronyms and other useful words or phrases. The key advantage of using the system is ease of maintenance. If the home page of our research group changes then the link can be updated in one simple action by altering the entry in the appropriate linkbase. The first goal of using the link service with AIMS is to author linkbases of official terms and definitions for the department with effort going towards automating link creation at every opportunity.
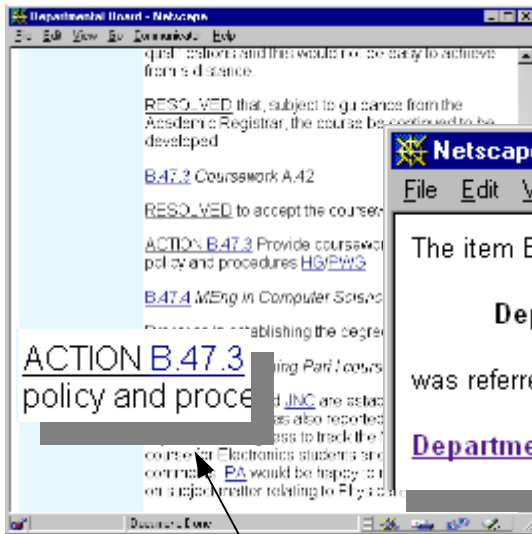
Linking Items in Minutes

A more advanced application of the linking service is to try and make more use of the structure of the minutes stored in AIMS. In the diagram below [Fig. 1] the Distributed Links Service has been modified in an experiment to link the items and actions in minutes together. The minutes of the Department Board of Electronics and Computer Science contain item numbers of the form B.n where n is a number. Frequently items in minutes will be following on from Actions and Items in previous editions of the minutes. The Distributed Links Service has been used to link together these item numbers enabling readers to follow the trails of issues back through time.

Item B.2 contains a reference to an ACTION B.47 in the previous set of minutes. The user follows the link B.47. This link triggers a program on the AIMS server.
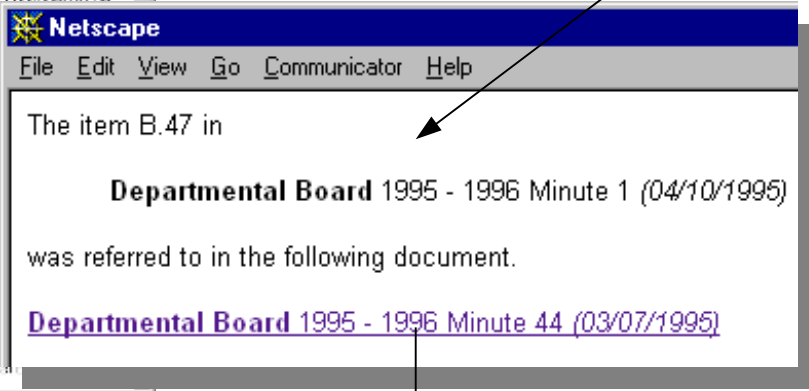
A program on the AIMS server searches the database to find the most likely document containing the item B.47 by using the date of the document the user is viewing. The result is below.

B.2 Actions

ACTION B.47 Draft Chair of Academic

The server returns a link to the referred document. In the body of this document is the Action B.47.3

Figure 1: Using the Distributed Links Service to link items in minutes allowing the user to follow trails of issues back through time.

This example needs further investigation to ensure that the techniques apply to more generic document types and are not relevant to just a few documents that match a certain pattern. For pattern matching algorithms to be more successful users must be trained to be accurate in their use of the numbering schema. Users submitting documents could be constrained in what they can enter in the fields or more fields could be added to describe the document in more detail. This will quickly discourage users especially if they lack confidence and are concerned that their categorisation would be considered 'incorrect'.

Link Authoring From A Browser

Webcosm ships with a Java link editor form. A more preferable solution would be to create links whilst viewing the documents. Microsoft Internet Explorer 4 allows users to add items to the context menu allowing the creation of a simple authoring tool. The user opens the destination and selects the appropriate word in the text to use as

the anchor. An additional entry on the context (or right click) menu 'Create Generic Link' uses Javascript to create a filled in form for submission directly to the Webcosm server.

This experimental application has enough functionality to allow users to directly author links between documents. The links will also have a high probability of still being relevant as new versions of documents are added. If the destination document is updated then the link will point to the new version of the document. If a document containing an anchor is updated then the link will survive if the same word or word-pair still occurs in the content of the new version. Whether these links are still relevant to the content is a more complicated issue. If the experiment is pursued further then this will need to be addressed.

Alternative Ways of Using the Link Service

Experience of the use of Webcosm and the DLS in a number of projects have shown that the implementation of the service as a Web proxy has a number of disadvantages. Users may not know how to set up a proxy or may not be allowed to make use of a proxy. The use of firewalls may restrict its operation. When in use the service will alter all documents the user reads no matter whether the links are in context or not. This particular issue has been addressed by the DLS which allows a linkbase to be used for a limited URL domain.

Alternative implementations do exist. Webcosm has also been developed as a plug-in library using the NSAPI interface for Apache Web servers. The link service is a library resident on the Web server and alters Web pages before the Web server delivers them to the browser. The consequence is that only documents on the server are enhanced. This is not a problem for AIMS and could be seen as a definite advantage. A similar low level interface to the Domino Web server does not currently exist though there are other methods for achieving a similar effect. Webcosm is implemented as a Web proxy component and a link server engine. It is possible to write programs that communicate directly with the link server. Within the AIMS application programs have been written that run every time a document is requested from the server. Java code asks the link server for the links for that particular document when required. The link service returns just the list of links it would have placed into the document and these links are processed and placed into a field in the document just before the Domino server renders and delivers the document. The links appear in the margin as a list of recommended sites for the user to view. Though the links are not being embedded in position in the main flow of the document the links are being generated on demand and in context depending on the users chosen linkbases.

The next step has been to try a process of pre-compilation of links. A system has been devised that can 'pull' the body of AIMS documents through the link service which adds the appropriate links. This HTML representation is stored in the database and served to users who are unaware that the main portion of the document is not actually being generated on demand. Though this has the obvious disadvantage that links are not being created on the fly the system now looks and feels like a normal Web site to users. This will make it easy to develop a system for creating point to point links between documents. The envisaged procedure would be for the link author to update documents overnight as required. This is still in development and no conclusions have yet to be drawn from its use.

Beyond Generic Links

The limitations of generic links soon become apparent. Too many links will overwhelm the user and be a distraction. This problem has been addressed by work done for the Open Journals project by modifying the DLS. The DLS has a concept of link priority schemes, different links can be given a priority by their author and the system can colour the links accordingly or even not display certain priority levels of links. Using pattern matching to create links in text is shown by the example of linking Board Minutes to be a promising way to find more link anchors with less explicit link authoring. Another application of such techniques would be to help with recognising names of people within documents though such techniques are much harder to implement. Pattern matching algorithms to solve this problem would need to be complicated compromising the performance of the service and slowing the access times of documents. There needs to be an alternative approach to the proxy synchronously processing each page. See [Carr et al 1998a]. The proxy could farm out the processing of documents to agents that would each examine the document and try to provide the user with useful information and links. These results would need to be delivered to the browser separately from the document causing a number of implementation problems..

An important implementation issue is how to create useful links for the documents. Some linkbases are automatically generated by extracting data from other online systems such as the staff database. This is useful but is limited to specific topics. Another method could be to require users to supply potential keywords and metadata for a document. This places unwanted demands on users and would be very difficult to implement. A more interesting method is to use some system that can understand the content of the documents and generate keywords automatically. Such a system, Refindment, has been developed by Multicosm and is being tested with AIMS.

Related Work

Other open hypermedia systems have evolved to encompass the Web in some respect and offer enhanced Web pages. Chimera [Anderson et al 1994] has been integrated with the Web in a number of ways [Anderson et al 1995]. One was to use a Java applet inserted into all Web pages that gave independent access to the information in the Chimera store. Another method was to write a CGI gateway from Web server to Chimera server allowing a browser access to the Chimera structured information. Hyperwave [Hyperwave 1999], now marketed as an advanced Web server, started life as an open hypermedia system Hyper-G [Andrews et al 1995], [Maurer 1996]. In it links are first class objects stored separately from documents and are also bi-directional. Links are merged into the documents when delivered to the browser. Webvise [Grønbæck et al 1999] has evolved from the DHM/WWW integration project into an enhanced Web service offering contexts, links, annotations and guided tours stored in external databases. Users can collaboratively create links and other hypermedia content using the Microsoft Internet Explorer 4 Browser utilising COM technology. The same group is now drawing on the lessons of many of the systems listed here and working on the ARAKNE framework [Bouvin 1999], [Bouvin 1999a]. This is an object-oriented component-based, three layer model aimed at providing Web augmentation tools a unified access to structure servers, proxies and Web browsers.

The Memoir [DeRoure 1998] project used an open architecture including proxies and link services to support researchers working with vast quantities of distributed information. The system not only used the DLS link services but also supported the trail as a first class object. Users could record their trails through documents whilst pursuing a particular task and the system could match trails thus connecting users with similar interests.

A number of systems have used Web server or proxy based methods to provide annotation services for applications such as discussion groups. Critsuite [Critsuite 1999] includes the CritLink link service component that will annotate existing documents with links. The author's first mediator was Shodouka [Shodouka 1999], this proxy could render Japanese Web pages as images on Web browsers that did not support the Japanese language.

The Wide Area Information Browsing Assistance (WAIBA) [Brooks et al 1995] project was set up to investigate the use of the Web for collaborative work. One of the underlying technologies developed was transducers [Meeks et al 1996]. Transducers were defined as proxies that alter the message streams between server and browser. A toolkit, STRAND (Stream TRANsDucer) was produced that allowed developers to rapidly build transducers. It was available until recently for free use as part of WebWare. Another architecture for building transducers or intermediaries is WBI [Barrett 1997]. This is a toolkit of Java classes that are currently in use for building a variety of applications. These include a personalisation proxy for a user to run on their own machine that will annotate pages the user reads. A more powerful system is to use a pair of transducers that can convert HTML or XML into other formats to transmit using wireless technology and then convert back at the client. A third use is as a filter for allowing children to use the Web.

Conclusion

We have created a Web based document management system using Lotus Domino with facilities designed for academic as opposed to a large-scale industrial environment. The use of such a flexible development platform now allows us to bring these documents alive for users in ways that are not possible on paper or on a traditional Web site.

All of this research is trying to find as many ways of adding value to the basic content submitted to AIMS. At its most basic AIMS simply stores files originally written on a word processor that were destined to be printed. These electronically frozen documents need to be kept alive by as many means possible to ensure their

continuing relevance to the users of the system and to make it easier for people to find the information they need on a daily basis. By making use of the content of these documents and many other live sources of information forms a considerable advantage over the paper versions outweighing the small amount of effort required to submit documents into AIMS.

References

[Anderson 1997] Anderson, K. (1997). Integrating Open Hypermedia Systems with the World Wide Web. *The Eighth ACM Conference on Hypertext and Hypermedia.*

[Anderson et al 1994] Anderson, K., Taylor, R. N., Whitehead, J. (1994). Chimera: Hypertext for heterogeneous software environments. *European Conference on Hypermedia Technology (ECHT94)*, pp. 97-107, Edinburgh, Scotland.

[Andrews et al 1995] Andrews, K., Kappe, F. and Maurer, H., (1995) Serving Information to the Web with Hyper-G. *The fourth World Wide Web conference.*

[Barrett 1997] Barrett, R., Maglio, P. P., Kellem, D. C. (1997). How to personalise the Web. *Conference on Human Factors in Computing (CHI97)*, Atlanta, Georgia, USA, Mar. 1997.

[Bouvin 1999] Bouvin, N. (1999). Unifying Strategies for Web Augmentation. *The Tenth ACM Conference on Hypertext and Hypermedia* .

[Bouvin 1999a] Bouvin, N., Schade, R. (1999). Integrating Temporal Media and Open Hypermedia on the World Wide Web. *The Eighth International World Wide Web Conference*, Toronto, Canada.

[Brooks et al 1995] 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, OReilly and Associates.

[Carr et al 1995] Carr, L., De Roure, D., Hall, W. and Hill, G. (1995). The Distributed Link Service: A Tool for Publishers, Authors and Readers. *The Web Journal* 1, 1, 647-656, OReilly and Associates.

[Carr et al 1998] Carr, L., De Roure, D., Hall, W. and Davis, H. (1998). Implementing an Open Link Service for the World Wide Web. *The World Wide Web Journal* 1, 2, 1998, OReilly and Associates.

[Carr et al 1998a] Carr, L., Hall, W., Hitchcock, S. (1998). Link Services or Link Agents *? Ninth ACM Conference on Hypertext and Hypermedia*.

[Critsuite 1999] Critsuite Critical Discussion Tools for the Web. The Foresight Institute. http://crit.org/http://crit.org/index.html.

[DeRoure 1998] DeRoure, D., Hall, W., Reich, S., Pirakis, A., Hill, G. and Stairmand, M. (1998). An Open Architecture for Supporting Collaboration on the Web. *IEEE Seventh Internation Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises.*

[Fountain et al 1990] Fountain A., Hall W., Heath I. and Davis H. (1990). Microcosm: an Open Model With Dynamic Linking. *Hypertext: Concepts, Systems and Applications (Proceedings of the European Conference on Hypertext*, INRIA, France, November 1990) 298 – 311.

[Grønbæck et al 1999] Grønbæck, K., Sloth, L., and Ørbæck, P. (1999). Webvise: Browser and proxy support for open hypermedia structuring mechanisms of the WWW. *The Eighth International World Wide Web Conference*, Toronto, Canada.

[Hill et al 1995] Hill, G., Hall, W., DeRoure, D. and Carr, L. (1995). Applying Open Hypertext Principles to the WWW. *International Workshop on Hypermedia Design*, Montpellier, France, 1-2 June 1995.

[Hitchcock et al 1997] Hitchcock, S., Carr, L., Harris, S., Hey, J. (1997). Citation linking: improving access to online journals. In *Proceedings of Second ACM Conference on Digital Libraries*, Philadelphia, pp.115-122.

[Hitchcock et al 1998] Hitchcock, S., Carr, L., Hall, W., Harris, S., Probets, S., Evans, D. and Brailsford, D. (1998). Linking electronic journals: Lessons from the Open Journal project. *D-Lib Magazine*, December 1998.

[Hyperwave 1999] Hyperwave Information Management server. Hyperwave GmbH. http://www.hyperwave.com/

[Lotus Notes 1999] Lotus Notes workflow and information management system. Lotus Development Corporation. http://www.notes.net/

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

[Meeks et al 1996] Meeks S., Brooks C. and Mazer M. (1996). Transducers and Associates: Circumventing Limitations of the World Wide Web. Paper presented at *etaCOM (Emerging Technologies and Applications in Communications) '96 conference*, Portland, Oregon.

[Shodouka 1999] Shodouka enables Japanese text to be read on any Web browser. The Shodouka Corporation. http://www.shodouka.com/

[Webcosm 1999] Webcosm is a component used in the knowledge management products produced by Multicosm Ltd. http://www.multicosm.com/