Link Augmentation: A Context-Based Approach to Support Adaptive Hypermedia

Christopher Bailey, Samhaa R. El-Beltagy and Wendy Hall

IAM Research Group
Department of Electronics and Computer Science
University of Southampton, UK
E-mail: {cpb99r,seb,wh}@ecs.soton.ac.uk

Abstract. In today's adaptive hypermedia systems, adaptivity is provided based on accumulative data gained from observing the user. User modelling, the capturing of information about the user such as their knowledge, tasks, attitudes, interests etc., is only a small part of the global context in which the user is working. At Southampton University we have formed a model of one particular aspect of context that can be applied in different ways to the problem of linking in context. This paper describes how that context model has been used to provide link augmentation. Link augmentation is an existing open hypermedia technique, which has a direct application in adaptive hypermedia systems. This paper presents a technique for cross-domain adaptive navigational support by combining link augmentation with a model of the user's spatial context.

1. Introduction

One of the main goals of any adaptive hypermedia (AH) system is to increase user efficiency. This efficiency is usually measured either in the time spent searching for information or increasing the amount of information absorbed by the user. The work presented in this paper is built around the philosophy of providing the user with greater access to information through link augmentation – a technique whereby external links are inserted directly into the body of a document. There are already several proxy-based systems that provide link augmentation such as Microcosm [9], Personal WebWatcher [18], and WBI [17], but they base their insertion algorithms on individual keywords or phrases in the document. However because the English language contains multiple uses for individual words, a simple augmentation algorithm like this can lead to out-of-place or irrelevant links and in addition to being frustrating, this can also lower a user's confidence in the system.

Out of place links are added when the component that adds those links fails to recognize a document's context. Such contextual information can be obtained by analysing the text in a document and comparing it against previously visited documents. This information can then act as a filter to remove or ignore those links that fail to match the current context.

While user modelling involves capturing some contextual information such as a user's knowledge in a particular area, their tasks, goals and interests etc., this information is often obtained using explicit feedback which can distract the user away from their original task [15]. One advantage of the technique used in this paper is the lack of explicit user feedback required. All information about the user is obtained implicitly from the user's trail and the contents of each page the user views. This removes the need to question the user and although not exploited in this paper, this data can be employed in other user modelling environments to infer details such as user interests, hobbies, skills and tasks.

Another advantage of this system is that it works without making any pre-defined assumptions about its users, thereby removing the need to bootstrap the system with user data. Additionally, since the trail capturing component is located on the end user's machine, adaptive link augmentation can also be provided across any hypermedia web page that the user visits.

This paper focuses on the extraction and analysis of a user's spatial context through a non-proxy agent-based architecture. Spatial context has been referred to elsewhere as the browsing context [16]. A method is presented for obtaining this information and using it as the basis of a linkbase-filtering algorithm. A linkbase is simply a database of links and the filter results in a single 'active' linkbase that contains a set of context dependent links. These links can be dynamically inserted into the current document. A new linkbase will be activated whenever the system determines that the user has entered a new context (providing the system has an associated linkbase). Link augmentation as a technique, has been shown to provide users with a viable means of decreasing search times [12], an combining this with the concept of spatial context, will benefit users and provide a new research area for adaptive systems.

2. Background

The role of linking has long been established in the hypermedia community where its primary use has been as a mechanism for navigation. Since the early days of adaptive hypermedia systems, links have been employed in many systems as a means of adaptive navigational support [5,10,18], and adaptive presentation [13].

The importance of links was reinforced in Brusilovsky's seminal 1996 paper [4] where he defined several subcategories of adaptive navigational support: direct guidance, link sorting, link hiding, link annotation and map adaptation. While today these categories remain more or less unchanged, it seems that an additional category can be added – (adaptive) link augmentation – which we define as the process of dynamically inserting additional links into existing web page. This differs from link annotation, which concerns the visible properties of hyperlinks, although these techniques can be combined to provide annotated, augmented links. The advantage of augmented linking is that the underlining navigational structure of the web page remains unaffected as all the original hyperlinks remain intact. However, the danger lies in information overload, which results when too many links are added, possibly leading to the situation where 'every word becomes a link'.

While there are several link augmentation systems, the earliest occurrence was seen in Microcosm [14,9], which was first developed in 1990 as a distributed open hypermedia environment that provided the user with dynamic, cross-application hyperlinks on the fly. These links were inserted (augmented) into the user's existing application and selecting one of these links issued a request to the Microcosm link service. This link service maintained a set of link databases (linkbases) and each link had one of three start point types: generic, specific or local. Specific links originate from an object at a specific point in the source document; local links originate from an object at any point in a specific document, and generic links, link from an object at any position in any document. Microcosm also provided text retrieval links where the user could highlight any text and ask the system to supply a set of related links.

One of the follow up projects to Microcosm, the DLS (Distributed Link Service) [7], is a link delivery system that operates in an open hypermedia environment. The DLS was aimed at bringing the concepts from the open hypermedia community to the Web. It acts as a link service providing other applications with hyperlinks on demand. These links are stored in multiple linkbases maintained by the DLS. However, by removing the need for hard-coded hyperlinks, the responsibility of determining link context fell on the shoulders of the user. So the major limitation of the system is in its inability to automatically switch between linkbases depending on the context of documents [11].

Today, many of the features found in both Microcosm and the DLS can be seen in Active Navigation's Portal Maximiser[1]. Essentially a web site server engine, Portal Maximiser provides many features such as document recommendations, contextual and relevance ranked search results, document categorization and theme-based dynamic (augmented) linking.

Another link augmentation system, WBI (WeB Intermediaries) [17], has been written as a web proxy that adds intermediary functions to the World Wide Web (WWW). WBI sits between the user and the outside WWW, analysing every page a user visits. It has a set of knowledge bases (KB's) hand authored for a specific subject. When the user views a related page, it replaces any known word or phrase with a hyperlink from the KB. If the user clicks on this hyperlink, a further information dialog box pops up on the client's machine offering additional resources (like word definitions or links to external web pages).

The PWW system [16] offers an approach to implicit (zero-input) personalization that is similar to the one taken in this paper. The system described by Kushmerick et al. is a server side system that offers recommendations to pages in the web site based on the URL and/or content of the referring page. User evaluations of PWW indicate the performance of the system reaches 9.3%, giving a value 77 times more effective then random guesswork. Our approach significantly differs from PWW by moving the architecture away from server side scripting thus allowing the system to gain a context of the user that extends across their entire browsing history and not just the referring page.

More recently, the Web, aided by improvements in browser technology, has seen the development of knowledge delivery systems that implement features similar to AH

[1] http://www.activenavigation.com/PortalMax/default.htm

systems. While such system lack any formal user modelling component, knowledge delivery systems, such as Flyswat[2], Zapper[3] and Atomica[4], provide resources such as link augmentation, keyword lookup, recommender functionality and shopping facilities to provide additional information to their users.

The current systems that employ link augmentation do so on the basis of the individual text of each word or phrase. If there is a match with a known link, then the word is replaced with the corresponding link. However this causes a problem when words have different meanings in different contexts. For example, the word 'java' may refer to the programming language Java, the country or the coffee bean and it is only by analysing the context in which the word is used that it is possible to distinguish between these meanings.

3. Context

Context is an important concept that has been examined in many different fields and for various tasks. It is also an involved issue, as it depends on the task at hand and the available variables that can be modelled in relation to that task. In the case of 'linking' the primary entities involved in this particular task, are those of the user and the document. There are many factors that can affect the context of the user. These include the user's role in an organisation/group/etc, their physical location, level of expertise in various topics, browsing history, interests, tasks, etc. Many of these user features are already captured in existing user models. The context of a document can be defined in many different ways such as by its content, its format (html, pdf, gif, etc), its purpose, the date it was created, the server on which it resides on, its download speed, etc. [11] and it's relationship with other documents. One particularly relevant system to the work presented in this paper and which has addressed the issue of 'linking in context' is the QuIC system.

3.1 QuIC

The work in this paper has drawn on work undertaken for a project called QuIC (Queries in Context). QuIC is a multi-agent system that was developed at the University of Southampton with the overall goal of utilizing concepts from the open hypermedia community to help users with their navigation and information finding activities. One of the issues addressed by the QuIC system is the use of linking in context as a way of assisting users in their information finding activities. This specifically targets a failing associated with traditional information retrieval models which is attributed to the isolation of these systems from the context in which queries are made [6].

[2] http://www.flyswat.com/

[3] http://www.zapper.com/

[4] http://www.atomica.com/

3.2 The QuIC Approach to Context

The model used by the QuIC project defines two factors for context: the interests of a user, and the contents of the document within which the links are to be rendered. A number of methods have been developed for using the content of unstructured information resources for inferring user interests for the purpose of constructing user or filtering models. In these models, the capture of user context or document context for the achievement of a specific task, is one of the goals. Depending on the specific task at hand, a number of techniques have been employed to build such models or profiles. Examples of techniques employed by Web agents to learn or capture a document or user profile include Decision trees, Neural Nets, Bayesian classifiers, Nearest Neighbour and TF-IDF (Term Frequency, Inverse Document Frequency) [19].

The decision was made to adopt a technique that would be capable of accurately capturing document context. TF-IDF is a very well studied and widely used information retrieval technique [22]. The technique is used to derive weights for terms in a way that would reflect their importance in a given document. TF-IDF is based on the vector space model where a vector is used to represent a document or a query. The cosine angle between different document vectors is a measure of how similar the documents are, and is used as a similarity function. Used in conjunction with a similarity function and other text processing techniques such as stop word removal and stemming, TF-IDF can be employed to distinguish between documents. The model has been used successfully for document ranking, document filtering, document clustering, and as the basis for relevance feedback [1]. One of the advantages of using the TF-IDF method is that unlike many other machine learning algorithms, it does not require large training data sets in order to distinguish between various documents. By representing a document through a vector space model computed via TF-IDF, comparing a document to other documents or queries is simply achieved through the application of a similarity function. The technique has therefore been employed by a large number of Web assistants, examples of which are: FAB [3], WebMate [8], and Margin Notes [21]. To further increase the accuracy of this method in distinguishing between different contexts, heuristics, as described in [12], have been introduced to the conditions of the context match to enable better determination of context.

The work resulting from the QuIC project has obvious applications for adaptive hypermedia systems, specifically adaptive navigational support such as link annotation and augmentation. To this end, the idea of context has been extended within this work to include the concept of a user's spatial context within an information domain.

4 Spatial Context

The goal of this work is to use the context technology described above to introduce the concept of spatial context into adaptive hypermedia systems. Such a system would work alongside existing user models providing another level of contextual information for the modelling component to draw upon.

A document's spatial context represents its location within the surrounding information domain. This spatial context is refined further by a user's path through the hyperspace before arriving at the current document. By doing this, a user is effectively selecting one path out of many other possibilities. In a hypermedia environment like the Web, the number of possible paths to a web page is continually changing, making it impossible for page designers to cater for the needs of all possible visitors. As a result, hard coded hyperlinks tend to cater for the 'average' user who has followed the most logical path to arrive at the current document.

4.1 Spatial Context in an Example Situation

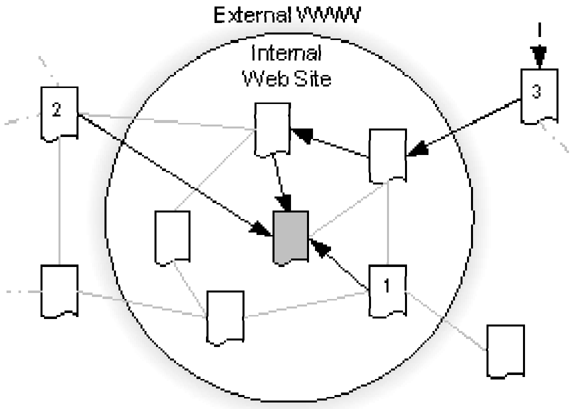


Fig. 1. Examples of a document's spatial context

Fig. 1 shows three separate paths (starting from documents labelled 1,2&3) taken to reach the central document (shaded grey). The black arrows represent the traversed navigational hyperlinks, while the grey lines represent alternative hyperlinks. In a real-world scenario, the central document might be a review of an XML book in an online bookstore such as Amazon[5]. Trail 1 would be a single link from one XML book review to other similar books. Trail 2 could represent a direct link from the author's external homepage to the current book, while trail 3 might be traversed by a user interested in Computer Science technologies, who visits the bookstore and searches through reviews of Network and Programming language books before finally arriving at the XML book review.

Each of these three trails represents a different context. If link augmentation were to be provided, each user would require a different set of hyperlinks. Situation 1 would require links to other XML books, the user in trail 2 should be presented with links to books by the same author, while in the third example, the augmented links should point the user to books on a wide range of computer science subjects. It is this issue of knowing when to activate these 'dynamic linkbases' that can be overcome by

[5] http://www.amazon.com/

understanding the user’s spatial context when arriving at the current document.

The following system uses the contextual component of QuIC to provide a means of capturing a user’s spatial context and, using this information, select and apply a dynamic linkbase.

5. System Architecture

The spatial context analyser has been implemented within an agent-based system built on top of the agent framework developed at Southampton University called SoFAR (Southampton Framework for Agent Research) [20]. SoFAR is a Java implemented framework designed as a test bed for agent research. SoFAR provides performatives for communication between agents, and ontologies for defining the contents of these communications. The decision for building an agent system arose due to the modular nature of the system’s components, which are well suited to an agent environment [2] and the desire to distribute the linking mechanism (which is essentially a DLS).

The network structure, as shown in Fig. 2, has been designed as a client-server approach. The agents (which run within the SoFAR framework) and user data all reside on a single server. In contrast to this, and in following with the architecture of the DLS, the linkbases used by the Linkbase Agent are fully distributed and can reside on any web server.

The client user interface is packaged as a downloadable Perl application and executed on the user’s machine. This interface communicates with the agent server through the use of sockets and also hooks into the client’s Internet Explorer Web browser through the Microsoft OLE (Object Linking and Embedding) automation feature that has been encapsulated as an ActiveX WebBrowser control. This allows the system to receive browser events such as OnLoad, DownloadComplete and DocumentComplete. However as a result, the system has been restricted to machines running Microsoft Windows with Internet Explorer 5.0 or later.

The system is triggered every time the Perl application receives a DocumentComplete event. The browser fires this event when it has finished loading both the text and images of a web page. For each new page the user views, the following steps are executed.

1. The Perl program first captures the URL of the current page and forwards this information onto the context agent.
2. An important issue for the context agent is that it needs to be aware that not every page is a possible candidate for context analysis. While some pages, such as Shockwave Flash sites, will produce empty context sets when processed which the system will ignore, others will produce misleading information, such as the ‘404 Not found’ and ‘301 Document Moved’ pages created automatically from broken hyperlinks. In order to identify these pages, the context agent employs a Single Layer Perceptron network (SLP) to extract the key features from the page and

apply a set of weights to these features. These features include identifying phrases like ‘Document Moved’, the amount of text in the page, the number of hyperlinks and the frequency of keywords like ‘broken’, ‘error’ and ‘404’. The SLP produces a probability, which determines its confidence in the page belonging to the ‘Page Not Found’ category. If this is high, the page is ignored; otherwise the context agent applies the TFIDF algorithm to the page and produces a ‘context model’ for this document.

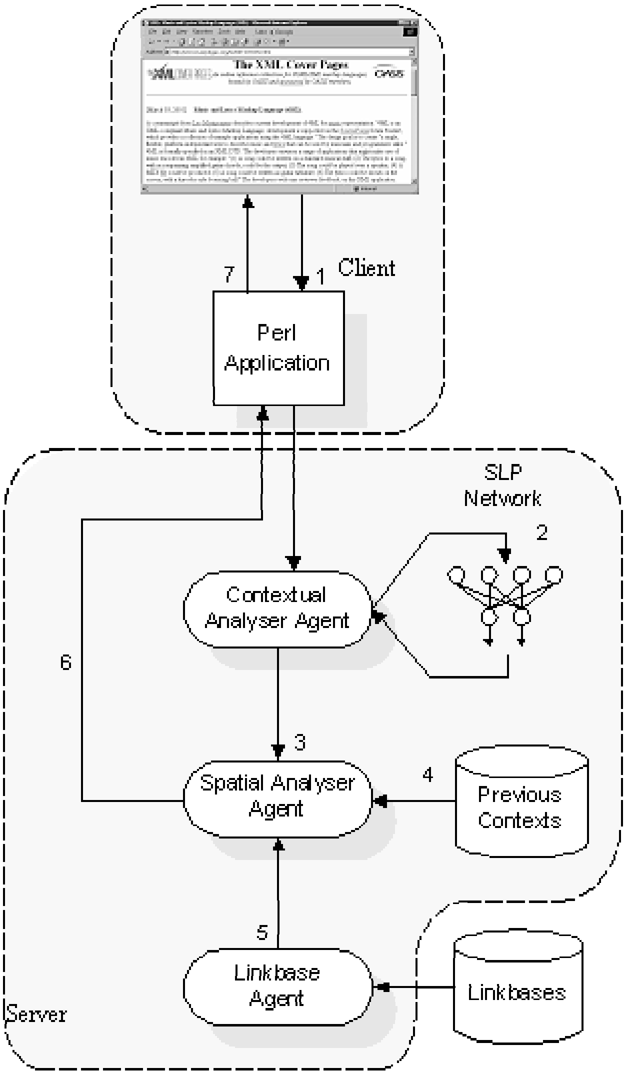


Fig. 2. The System Architecture

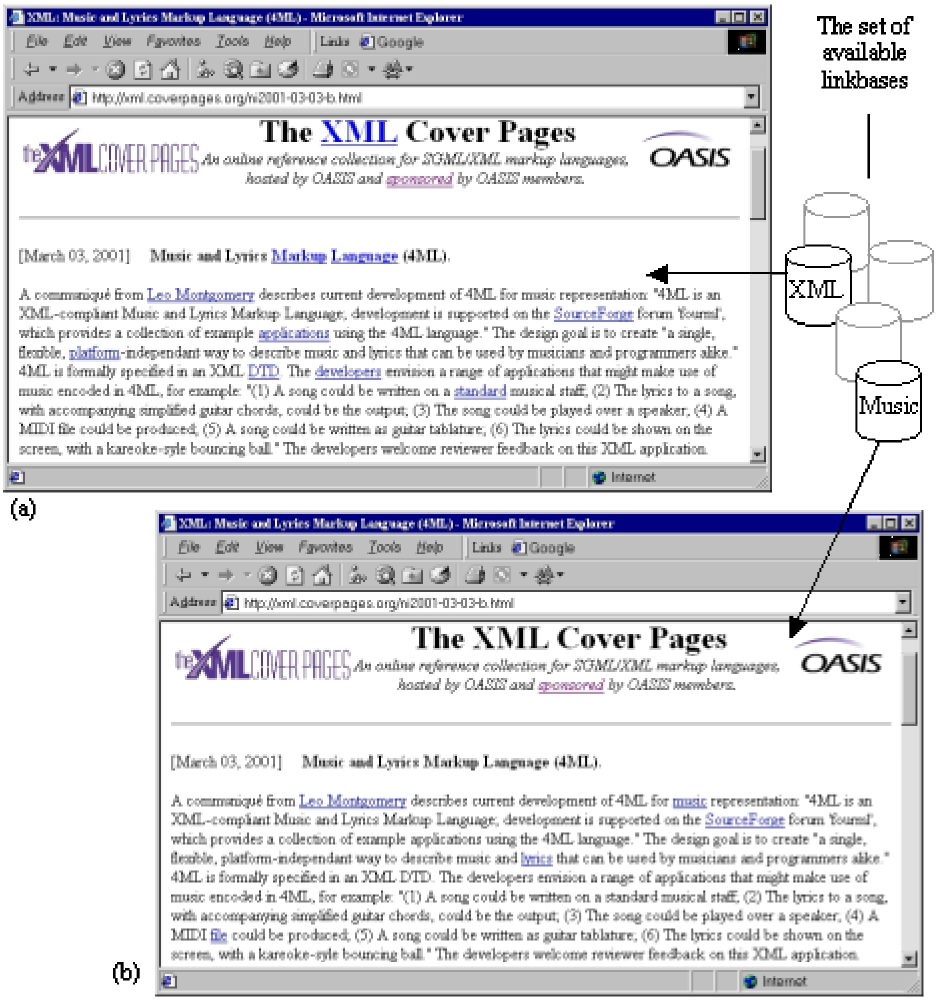
3. This model is then passed to the spatial context analyser to determine the context of the current user. Here there are three possible outcomes: the user is in the existing context, the user has returned to a previous context or the user has entered a new context. Firstly, this agent compares the new context model against the current model using a cosine similarity function. If a match is found, then the system moves onto step 5, otherwise step 4 occurs.
4. The spatial analyser agent compares the current context model against the entire set of previous context's that the user has experienced in the current browsing session. This comparison uses the same similarity function. If the highest match exceeds a given threshold then the system assumes that the user has returned to a previous context. This could occur for a number of reasons. For instance, the user could press the back button after arriving at an irrelevant page or the user has finished following a search thread and is returning to an old topic, this could also occur if the user has several browser windows open and is switching between them. If no match is found, then the current document's context forms the start of the user's new browsing context (and a record of it is stored in the 'Previous Contexts' database).
5. When the system has calculated the current context of the user, a request is sent to the linkbase agent for matching linkbases. This agent searches through each known linkbase to find the highest similarity match with the context. The linkbase agent returns the highest matching linkbase (or null if no match is found).
6. The system passes all the links from the matching linkbase onto the Perl application.
7. The last job of the Perl program is to extract the text from the web, page search through it and replace all the matching words with a hyperlink. Matches are found through simple string/word comparison. The text extraction is achieved through an ActiveX call to the browser's Document Object Model requesting all the text in the current web page. This request ignores the page type giving the link augmentation process the ability to operate on any type of web page including dynamically generated pages such as ASP, JS, CGI, PHP and SHTML. The resulting augmented page is reinserted back into the browser window using the properties of dynamic HTML.

This whole process is executed in real time and the user will see the requested web page displayed, and then a split-second later, the new links will appear. Because the page is held in memory, there is no visible refresh; all that is apparent is that certain words are instantly transformed into hyperlinks. Often there is no visible delay between loading the page and inserting the links.

This context-based technique works due to the property that all the augmented links reside in the same linkbase and each linkbase has been hand-crafted to contain relevant links on specific subjects. For instance, a linkbase on the subject of XML might contain individual links to:

?? W3C XML specification
?? XML parsers
?? XML technologies

The authors of these linkbases have the freedom to make each linkbase as detailed as desired, so for example, there could be separate linkbases for each of the above sub-topics or even linkbases of each sub-sub-topic.


XML: Music and Lyrics Markup Language (4ML) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Google
Address http://xml.coverpages.org/ni2001-03-03-b.html
The XML Cover Pages
theXMLCOVERPAGES
OASIS
Internet
The set of available linkbases
XML
Music
(a)
(b)

An online reference collection for SGML/XML markup languages, hosted by OASIS and sponsored by OASIS members.

[March 03, 2001] Music and Lyrics Markup Language (4ML).

A communiqué from Leo Montgomery describes current development of 4ML for music representation: "4ML is an XML-compliant Music and Lyrics Markup Language, development is supported on the SourceForge forum 'fourml', which provides a collection of example applications using the 4ML language." The design goal is to create "a single, flexible, platform-independant way to describe music and lyrics that can be used by musicians and programmers alike." 4ML is formally specified in an XML DTD. The developers envision a range of applications that might make use of music encoded in 4ML, for example: "(1) A song could be written on a standard musical staff; (2) The lyrics to a song, with accompanying simplified guitar chords, could be the output; (3) The song could be played over a speaker; (4) A MIDI file could be produced; (5) A song could be written as guitar tablature; (6) The lyrics could be shown on the screen, with a kareoke-syle bouncing ball." The developers welcome reviewer feedback on this XML application.

Fig. 3. An example link augmented page as viewed from an ‘XML’ and a ‘Music’ context

Fig. 3 shows the system in operation. It shows the same page is viewed from two different spatial contexts[6]. Page (a) appears as viewed by a user who has been reading articles about XML before arriving at the current document. In this instance, the system has determined that the user's spatial context best matches the 'XML' linkbase and so links from the keywords 'XML', 'DTD', 'standard' and 'developer' have been inserted. Although it appears that this page exists in a predominately XML context, the user viewing page (b) has previously been looking at music related sites and therefore their spatial context best matches the 'Music' linkbase and so the page has been augmented with the links 'lyrics' and 'music'.

6. Linkbases and Cross-Domain Support

By abstracting the links and storing them in a set of linkbase, the system gains cross-domain support 'for free'. This one-size-fits-all approach will provide link augmentation to any information domain contained within the linkbases. However the system relies heavily on both the quality and quantity of these linkbases. Absent linkbases will simply lead to a lack of augmented links for that domain, however badly authored linkbases can lead to unhelpful or irrelevant links. While the current linkbases have all been hand authored, it is desirable to find an automated link extraction algorithm that could be used to create linkbases to cover a variety of topics.

7. Future Work

The work introduced in this paper describes one way of capturing the user's spatial context within an hypermedia system like the WWW, and then applying hyperlinks based on this context. However there are several major hurdles that need to be overcome before such a system can have any practical benefits. Firstly, the system relies heavily on hand-authored linkbases and without these the system is useless. This can be overcome with a link generation mechanism as used in both Portal Maximiser and QuIC. Providing the system could produce links of a high enough quality, this would remove the need for human 'experts' that would currently have to spend many man-hours manually creating each linkbase.

A second issue that arises is the question of multiple active linkbases. Currently, only one linkbase is ever 'activated' at any one time. However it is often possible for the user to exist in several different contexts simultaneously and in such a situation it would be beneficial to activate multiple linkbases. If this is the case, then care must be taken to avoid causing the user to suffer from information overload and as a result, further investigation is needed to see whether there is such a demand for this service.

One important area for further research would be to investigate the level of intelligence required when inserting links into a document. The current link augmentation

[6] In these screen shots, the original document contains the hyperlinks: 'sponsored', 'Leo Montgomery' and 'SoundForge'.

algorithm matches document keywords words against the set of context-filtered links. If a match exists then the corresponding link is inserted. This can cause problems due to the disregard of the surrounding text - the paragraph context of the word. This is part of a bigger issue that concerns contextual scooping. Scooping examines the level at which contextual analysis is conducted. By analysing not only the document but also each paragraph and sentence, link insertion could become even more accurate.

Finally, while the system does indeed produce real-time context-dependent augmented links, there has yet to be any formal evaluation of the system. When the improvements stated above have been introduced, the final stage will need to involve a system evaluation.

8. Conclusions

While link augmentation is not a new technique and has been used in many systems before, applying context analysis as a means of filtering out irrelevant links is entirely new. The work here shows that link augmentation, when applied to a document's spatial context, is a highly significant area for further exploratory study. To support this claim, user evaluations of the QuIC project (as reported in [12]) and PWW [16] already show that link augmentation and recommendation are viable means of enriching the existing hypermedia domain with effective user-centric information which can be used as a means of decreasing search times.

The flexible design of the system allows link augmentation to be provided across a variety of information domains, dependent only on the availability of linkbases. In addition to this, the authors feel that adaptive link augmentation, when implemented with care, is a worthy addition to Peter Brusilovsky's list of adaptive navigation technologies and warrants further research.

9. Acknowledgements

The work presented in this paper has been supported by the QuIC project, ESPRC grant GR/M77086

References

1. Baeza-Yates, R. and Ribeiro-Neto, B. (1999). "*Modern Information Retrieval*". Addison-Wesley/ACM Press.
2. Bailey, C and Hall, W. (2000) "*An Agent-Based approach to Adaptive Hypermedia using a link service*". P. Brusilovsky, O. Stock, C. Strapparava (Eds.): Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000, pp. 260-263

3. Balabanovic, M. and Shoham, Y. (1997). *"Fab: content-based, collaborative recommendation"*. Communications of the ACM, 40(3), pp. 66-72.
4. Brusilovsky, P. (1996). *"Methods and Techniques of Adaptive Hypermedia"*. Journal of User Modelling and User-Adaptive Interaction, UMUAI6.
5. Brusilovsky, P., Eklund, J. and Schwarz, E. (1998). *"Web-based education for all: A tool for developing adaptive courseware"*. Computer Networks and ISDN Systems. Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998, 30 (1-7), pp. 291-300.
6. Budzik, J. (2000). *"User Interactions with Everyday Applications as Context for Just-in-time information Access"*. In Proceedings of Intelligent User Interfaces (IUI), ACM, New Orleans, LA USA, pp. 44-51.
7. Carr, L., De Roure, D., Hall, W. and Hill, G. (1995). *"The Distributed Link Service: A Tool for Publishers, Authors and Readers"*. World Wide Web Journal 1(1), O'Reilly & Associates (1995). pp 647-656.
8. Chen, L. and Sycara, K. (1998). *"Webmate: A Personal Agent for Browsing and Searching"*. In Proceedings of the Second International Conference on Autonomous Agents, ACM, Minneapolis, pp. 132-139.
9. Davis, H.C., Hall, W., Heath, I., Hill, G. and Wilkins, R.J. (1992*) "Towards an Integrated Information Environment with Open Hypermedia Systems"* In Proceedings of ECHT'92, ACM Press, pp. 181 – 190.
10. De Bra, P. and Calvi, L. (1998). *"AHA: a Generic Adaptive Hypermedia System"*. In Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, HYPERTEXT'98, Pittsburgh, USA, June 20-24, 1998.
11. El-Beltagy, S. (2001). "An Agent Based Framework for Navigation Assistance and Information Finding in Context". PhD thesis. University of Southampton, 2001.
12. El-Beltagy, S., Hall, W., De Roure, D. and Carr, L. (2001). *"Linking in Context"*. To appear in the Twelfth ACM Conference on Hypertext and Hypermedia (HT'01), Denmark, 2001.
13. Espinoza, F. and Höök, K. (1997). *"A WWW Interface to an Adaptive Hypermedia System"*. Presented at PAAM (Practical Applications of Agent Methodology), April 1996, London. http://www.sics.se/~espinoza/pages/PAAM_submission.html
14. Fountain, A., Hall, W., Heath, I. and Davis, H. C. (1990) *"Microcosm: an open model with dynamic linking"*. In Hypertext: Concepts, Systems and Applications (A. Rizk, N. Strietz, and J. Andre, eds.), (France), pp. 298-311, European Conference on Hypertext, INRIA, November 1990.
15. Kim, J., Oard, D. W., and Romanik, K. (2000) *"Using implicit feedback for user modeling in Internet and Intranet searching"*. Technical Report, College of Library and Information Services, University of Marylandat College Park.
16. Kushmerick, N., McKee, J. and Toolan, F. (2000). "*Towards Zero-Input Personalization: Referrer-Based Page Prediction*". P. Brusilovsky, O. Stock, C. Strapparava (Eds.): Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000, pp. 133-143
17. Maglio, P., P. and Farrell, S. (2000). *"LiveInfo: Adapting web experience by customization and annotation"*. In Proceedings of the First International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2000). LNCS Series, Springer-Verlag.

18. Mladenic, D., (1996). *"Personal WebWatcher: Implementation and Design"*. Technical Report IJS-DP-7472, Department of Intelligent Systems, J.Stefan Institute, Slovenia.
19. Mladenic, D. (1999). *"Text-Learning and Related Intelligent Agents: A Survey"*. IEEE Intelligent Systems, 14(4), pp. 44-54.
20. Moreau, L., Gibbins, N., De Roure, D., El-Beltagy, S., Hall, W., Hughes, G., Joyce, D., Kim, S., Michaelides, D., Millard, D., Reich, S., Tansley, R. and Weal, M. (2000). *"An Agent Framework for Distributed Information Management"*. In The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, Manchester, UK.
21. Rhodes, B. J. (2000) *"Margin Notes: Building a Contextually Aware Associative Memory"*. In Proceedings of Intelligent User Interfaces (IUI '00), ACM, New Orleans, LA USA, pp. 219-224.
22. Salton, G. and McGill, M. J. (1983). *"Introduction to Modern Information Retrieval"*. McGraw Hill, New York.