Andreas Heuer, Ernst-Georg Haffner, Uwe Roth, Christoph Meinel:
A Hyperlink Focused Browse Assistent for the World Wide Web
in Proceedings of the 1st International Conference on Internet Computing (IC 2000), CSREA Press, Las Vegas, USA, pp. 79-84, 6, 2000.
ISBN: 1-892512-65-3.

A Hyperlink Focused Browse Assistant for the World Wide Web[*]

Andreas Heuer
Institute of Telematics
Bahnhofstraße 30-32
D-54292 Trier
Germany

Ernst-Georg Haffner
Institute of Telematics
Bahnhofstraße 30-32
D-54292 Trier
Germany

Uwe Roth
Institute of Telematics
Bahnhofstraße 30-32
D-54292 Trier
Germany

Christoph Meinel
Institute of Telematics
Bahnhofstraße 30-32
D-54292 Trier
Germany

Abstract *This paper describes a browse assistant focusing on hyperlinks. It discusses the concept and an accompanying prototype implementation of the assistant. The aim of the assistant is to increase the usability of navigation through the World Wide Web (WWW) by the provision of more detailed hyperlink information for each browsed HTML-document. Extracted from a personal link database it offers helpful background information for each referenced document. Therefore the tool supports navigation and allows a brief classification of hyperlinks based on the given additional information.*

Keywords: browse assistant, usability, link database, link management, Java Application

1 Introduction

Given the enormous growth of the WWW, the question of its usability arises. It becomes more and more difficult to find relevant resources in the ever growing mass of online available documents. Therefore navigation strategies are a very important basis for successful browsing sessions. Since navigation in the WWW is based on hyperlinks one starting point for an improvement of the current situation is obviously their personal management. As described in [Procter99] the present generation of web-browsers unfortunately provides the user with too little information to help them decide whether to choose a link. Each navigation decision of the user is based on the available URIs and their link representation in the documents. Implicitly given information, such as the document-type, combined with some previously achieved experiences lead to a decision to select a link.

While Procter proposes a link-decision support by a kind of retrieval of meta data for given URIs, Chakrabarti [Chakrabarti99] goes even a step further and enhances the documents content itself with reverse citations, so called back-link information, that is also available via HTTP-extensions. These examples show that, although there are some very helpful features already implemented in the current browser generation (i.e. the "history-function" in IE5 is really nice), there are still requirements for assistant tools that improve the usability further. As mentioned above, one basis for improvements in user-support is a qualified hyperlink management, based on a link database. In spite of other link databases [Pitkow96] that were mainly designed to maintain link consistency, our approach focuses on a personal link database, that stores information about the link-structure of the browsed documents. While no automated content-based bookmark generation (compare [Maarek96]) is immediately planned, future moves in this direction, based on case based learning are thoroughly on our mind.

Our vision when creating the assistant was an additional window visible during the browsing session on demand, that gives an overview about link-relations of the document actually displayed in the web-browser. The window displays a link-list that have their origin in this document and allows to display them in the browser by a double click. Consequently, users will get a very quick, but nevertheless very informative,

[*] As in Proceedings of the 1st International Conference on Internet Computing (IC'2000), Las Vegas, Nevada, USA, 2000

overview about the references of the document. Obviously the result is quite similar to the reference-list in scientific publications. It can be very helpful for getting a first impression about the relevance of the document. Of course, further information can be attached to each listed URI, helping the user to classify the link as well as the document itself. Such additional information allowing a classification could be the number of times the linked URIs have been visited previously or the last date the URIs have been requested. Also, the number of times a certain document served by the web-server was visited could be helpful as well as a short comment about the web-site available from a previous visit made by the user himself/herself.

2 Technical Concept

The functionality of the browse-assistant splits up into on three independent areas. These are the filtering and extraction of the data, the storage, and finally the visualization. Given a conventional setup, where the client/browser connects to the internet directly or via a proxy, the first component of the assistant tool has to be located between the client and the internet, respectively the proxy. This component is required to gather the data, since the assistant tool is not integrated into the browser at the moment. In this setup this component of the assistant works as a gateway transferring data between the client and the server. Furthermore, it processes each request of the browser and the corresponding reply of the server but must not restrain users at their browsing activity. Processing request and reply is a task that splits up into several sub-tasks. At first, the request has to be parsed so that the requested URI is available for the system. Second, this URI has to be stored with a timestamp indicating the date and the time the request was made. Third, the reply of the server is being classified. If the reply has an OK status and the content is a HTML-document, the data will get parsed. Hyperlinks are being extracted. Referenced URIs as well as the link-texts (link-text is the text between the anchor tags, <a href=...>link-text</a>) will be available after this step.

This is where the second component, the storage component, gets involved. In this concept, a database will be employed to store the available data. A database is used instead of something like a flat file for several reasons: First, a database is designed to match structured data. Second, it provides features for data management as well as for data retrieval. Third, a database can easily handle the amount of data that will be accumulated during extensive use of the assistant tool. Nevertheless mechanisms have to be implemented that keep the database consistent but remove unnecessary contents.

The last component employed in the concept is the visualization component. This component provides the user-interface for the assistant tool. It displays navigational information about the current document based on the URI of the request and the data parsed from it to the user. On the one hand, the tool lists the links that have their origin on the current document. On the other hand, previously browsed documents that refer to the current document will be listed. The concept anticipates several extensions to these lists. The lists have to show additional information for the corresponding URIs of the links. At first it should show if the URI was visited in an earlier session. If so, the last access-time and the number of overall visits of the corresponding URI should be displayed. Furthermore, comments on the URI provided by the user himself in previous sessions can be visualized, if required. Also, several algorithms can be implemented to order the URIs contained in the list as well as a filtering on criteria chosen by the user. Such criteria could be the document type (image, text/html), or the occurrence of certain hosts, etc.

The combination of the three described components results in an browse assistant that enables the user to improve his/her navigation through the ever increasing space of the WWW. Furthermore each component on its own can be extended easily to provide additional functionality. Such features can be a more detailed history list [Heuer99] or an local search engine [Heuer99a] based on the browse history.

3 Prototype

In order to provide a highly browser- and platform-independent tool, the prototype implementation was done in JAVA with respect to the gateway and the visualization component. While the gateway runs as a JAVA-application, the assistant itself can either be run as an application or as an applet. In both cases, gateway and assistant, a quite satisfying platform-independence is being guaranteed. For data storage, a conventional relational database is used that is accessed via JDBC [Reese97] from both the gateway and the visualization component. Therefore, any relational database product for which a JDBC-Driver is available and which implements the SQL-standard [Date 97] may be chosen.

3.1 The Gateway

The prototype implementation consists of a multi-threaded server program [Sridharan97] that transfers each HTTP- request of the browser to the appropriate web-server. Since this component is designed to gather data about the browsing sessions of the user, it processes each request. While each reply is directly forwarded to the browser and logged with the accompanying HTTP-status in the data storage component, the processing of the data is dependent on the document type. Concerning the field of application, the extraction of hyperlinks, only the HTML-document-format is relevant for parsing. An HTML-parser, designed to understand various HTML dialects, extracts all links from the document and fully qualifies the URIs. In a second step the extracted data (the referenced URIs) is stored in a suitable way in the storage component.

3.2 The Database

For the use of the prototype implementation, a relational database was chosen. The assigned schematic data model consists of four entities [Figure 1], the URI, the Visit, the Content (the document) and the Link. The URI entity consists of the URI-String in a non-relative, fully qualified, absolute form, and a comment-string for any remark of the user that refers to the URL. The Visit is a timestamp, combined with the HTTP-status that indicates the time at which a URL was requested and the reply status. Obviously, an URI can be visited more than once, so a relation of "one to n" between the URL and the Visit exists. Since the document belonging to an URI may change in time, the entity Content was designed. It stores document-relevant data such as the content-length, the content type and a timestamp indicating when this content was found at the given URI. Therefore, this entity has a relation of one-to-n to the URI entity. At last, the fourth entity, the Link, connects the Content-entity with the URI-entity. The Link-entity contains the link-text as well as the referring document and the linked URI. Therefore, a relation of one-to-n exists between Link and URI and between Link and Content. This model covers the functionality provided by the assistant tool and can be easily extended for further use. Since performance is a very important for any assistant tool, the creation of indices on certain table columns in the model is recommended.

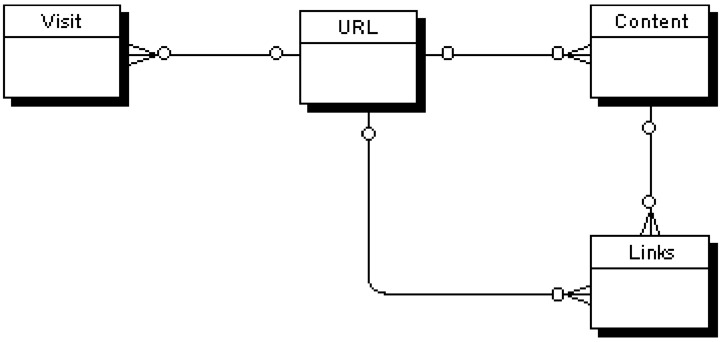
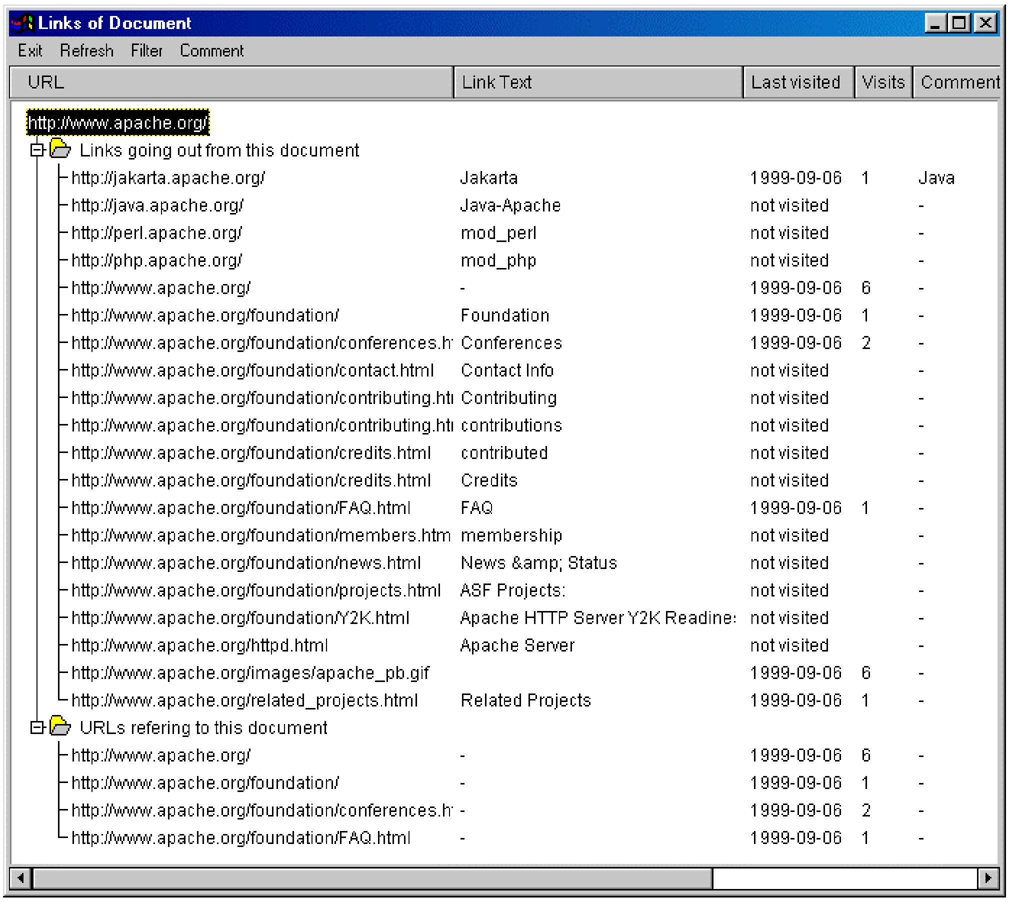


Figure 1: A schematic overview of the entities employed by the prototype implementation

3.3 The Visualization Component

The visualization component provides the user interface of the browse-assistant. It consists of a separate frame that is opened either by the applet or application of the assistant tool. The frame displays a tree-view, compare [Czerwinsky97]. The current document is represented by the root-node of the tree-view. It contains at the two separate folders and can be extended for an arbitrary number of further folders in future. The first folder contains the URIs of those links that have their origin at the current document as they have been extracted by the gateway. The second folder contains the URIs of previously visited documents that refer to the current document.



URL	Link Text	Last visited	Visits	Comment
http://www.apache.org/				
Links going out from this document				
http://jakarta.apache.org/	Jakarta	1999-09-06	1	Java
http://java.apache.org/	Java-Apache	not visited		-
http://perl.apache.org/	mod_perl	not visited		-
http://php.apache.org/	mod_php	not visited		-
http://www.apache.org/	-	1999-09-06	6	-
http://www.apache.org/foundation/	Foundation	1999-09-06	1	-
http://www.apache.org/foundation/conferences.h	Conferences	1999-09-06	2	-
http://www.apache.org/foundation/contact.html	Contact Info	not visited		-
http://www.apache.org/foundation/contributing.ht	Contributing	not visited		-
http://www.apache.org/foundation/contributing.ht	contributions	not visited		-
http://www.apache.org/foundation/credits.html	contributed	not visited		-
http://www.apache.org/foundation/credits.html	Credits	not visited		-
http://www.apache.org/foundation/FAQ.html	FAQ	1999-09-06	1	-
http://www.apache.org/foundation/members.htm	membership	not visited		-
http://www.apache.org/foundation/news.html	News &amp; Status	not visited		-
http://www.apache.org/foundation/projects.html	ASF Projects:	not visited		-
http://www.apache.org/foundation/Y2K.html	Apache HTTP Server Y2K Readine	not visited		-
http://www.apache.org/httpd.html	Apache Server	not visited		-
http://www.apache.org/images/apache_pb.gif		1999-09-06	6	-
http://www.apache.org/related_projects.html	Related Projects	1999-09-06	1	-
URLs refering to this document				
http://www.apache.org/	-	1999-09-06	6	-
http://www.apache.org/foundation/	-	1999-09-06	1	-
http://www.apache.org/foundation/conferences.h	-	1999-09-06	2	-
http://www.apache.org/foundation/FAQ.html	-	1999-09-06	1	-

Figure 2: A screenshot of the prototypes implementation of the visualization component used in an example case. While the root nodes name is the current URI, the first folder contains the links that start at the current page. In spite of that the second folder contains the URIs of documents, that have been browsed previously and refer to the current document. The data attached to the nodes, respectively the links, are the URL, the link-text , the last visit, the number of times the URL was visited and a comment.

The employed tree-view-component allows to attach any given data to the displayed nodes [Figure 2]. In our case this data is the additional information provided for the listed URIs. The number of times an URI was visited already belongs to this category as well as any comment and the date of the last visit.

The menu on the top of the frame allows to set filter criteria for the displayed URIs, respectively documents. Available options are the content-type, the timestamp indicating the last access of the document and a list of hosts. Filtering of URIs by content-type allows to display only certain document formats. Since it may be necessary to reduce the number of referencing documents it is furthermore possible to show only referencing documents that have been visited in a given time-frame. Although the browse assistant may not be all that useful at very commercial sites, suppressing several hosts in the list of the displayed URIs allows to hide links that belong to ads or counters.

Finally, a comment that will be inserted into the database can be set for the actual document, as well as for any displayed URI. It will be displayed the next time the URI is listed as a node in the first or second folder. If the visualization component is executed as an applet, each node can be opened in a new browser window easily. If the visualization component is executed as an application, a new process has to be started, parameterizing the browser with the URL.

3.4 Problems and Limitations

Of course there are several problems that evolve with the use of the browse-assistant prototype. The visualization component depends on the data provided by the gateway. Since a browser holds its own cache, the gateway does not always know which document is actually being displayed by the browser. Furthermore, the user may navigate to a document by using the back-button of the browser. Since the visualization component is unable to determine which document is being displayed, it cannot become synchronized with the browser anymore. This asynchronous behavior will cease as soon as the browser again requests a document via the gateway. In the mean time manual synchronization by forcing the browser to reload a document (e.g. shift+reload) again is required.

As discussed in the outlook the high amount of data to be handled by the system may reduce the performance dramatically if there is no "clean-up" mechanism provided.

4 Outlook

Several issues remain to be discussed about the future development of the prototype. Such issues are the provision of an offline evaluation-interface for the database, a comfortable cleaning and validation mechanism and a graphical configuration interface for the gateway.

Obviously, with the continual use of the tool, the database becomes valuable in time. The data stored inside the database can be very interesting for further "offline" evaluations. An evaluation-interface would allow the user to get answers to questions like: "When was that document ever browsed?", "When did I browse the document for the fist time?", "How often did I visit that web-site?", etc. Supposing that comments will be used intensively and that the ranking of URLs will be allowed, we can infer that something like a history-based bookmark-file might, probably, evolve.

Of course, the value of the database depends on how well the user takes care of the management of the data stored inside. This is why a need for an-easy-to-use, fast tool arises that enables the user to "clean up" the database. Naturally, the "cleaning up" can also be done before the data is stored in the database. A comprehensive configuration of the gateway during the browsing process would guarantee the quality of the links stored in the database and reduce the number of "useless" URIs dramatically. Supposing that the user visits the same sites [Tauscher97, statmarket] very often, the removal of such URIs would be very effective.

5 Conclusion

The concept of the browse-assistant described in this paper allows fast transit from one document to references inside of it. Obviously, the additional information provided for the links contained in the document helps to decide which link to follow in his browsing session. Furthermore information about the previously browsed documents that refer to the current document is very valuable. On the one hand, it gives a good overview about related documents, on the other hand it allows quick transit to these documents by double clicking the URI. By employing the three-component-model presented in this paper, an extensible implementation of the tool can be carried out completely platform- and browser independently. This way, a broad field of employment for the tool can be guarantied.

6 Reference

[Czerwinsky97] Czerwinsky, M., & Larson,K. (1997), *An Initial Examination of Tree*

Navigation versus Hyperbolic Browsing during Search, The Microsoft Site Analyst. British HCI, http://www.research.microsoft.com/research/ui/marycs/bhci97.html

[Chakrabarti99] Soumen Chakrabarti, David A. Gibson, Kevin S. McCurley, *Surfing the Web Backwards*, 8th International World Wide Web Conference, Toronto, Canada, May 11-14, 1999, online available http://www8.org/w8-papers/5b-hypertext-media/surfing/surfing.html

[Date 97] C.J. Date and H. Darwen, *A Guide to SQL-Standard*, Reading, MA: Addison – Wessley 1997

[Heuer99] Andreas Heuer, Christoph Meinel, *Database based Navigation Assistant*, in Proceedings of WebNet 99- World Conference on the WWW and Internet, Honolulu, Hawaii, October 24-30, 1999, pp. 505-510

[Heuer99a] Andreas Heuer, Christoph Meinel, *Database based History Browse Assistant*, in Proceedings of Multimedia Systems and Applications (IMSA'99), October 18-21, 1999, Nassau, the Bahamas, (ISBN 0-88986-271-0) pp. 78-83

[Maarek96] Yoelle S. Maarek, Israel Z. Ben Shaul, *Automatically Organizing Bookmarks per Contents*, Fifth International World Wide Web Conference, May 6-10, 1996, Paris, France, online available at: http://www5conf.inria.fr/fich_html/papers/P37/Overview.html

[Reese97] Reese, G. (1997). *Database Programming with JDBC and Java*, O’Reilly

[Pitkow96] James E. Pitkow, R. Kipp Jones, *Supporting the Web: A distributed Hyperlink Database System*, Fifth International World Wide Web Conference, May 6-10, 1996, Paris, France, online available at: http://www5conf.inria.fr/fich_html/papers/P10/Overview.html

[Procter99] Rob Procter, *Improving Web Usability with the Link Lens*, 8th International World Wide Web Conference, Toronto, Canada, May 11-14, 1999, online available http://www8.org/w8-papers/4b-links/improve/index.html

[Sridharan97] Sridharan, P. (1997), *Advanced Java Networking*, Prentice Hall

[Tauscher97] Tauscher, L., & Greenberg, S. (1997), *How People revisit web pages: empirical findings and implications for the design of history systems*, International Journal of Human Computer Studies, 1 (47), 97-137 http://www.hbuk.co.uk/ap/ijhcs/webusablity/tauscher/tauscher.html

[statmarket] *Loyalty Index (Repeat-Visitors)*, http://www.statmarket.com/SM?c=Loyalty_Index