Navigation-Dependent Visualization of Distributed Internet Structures

Dirk Kukulenz, Josef Pauli
Institute of Computer Science and Applied Mathematics
Preusserstr. 1-9, 24105 Kiel, Germany
dku@ks.informatik.uni-kiel.de

Abstract

It is one main aspect of the internet to be a distributed, multimedia database offering a fast growing amount of valuable information. Search in this database can be done by posing questions to search machines like Alta Vista. On the other hand it is possible to browse through the space of connected information objects. When browsing the problem arises that the user usually has no special knowledge of the structure of the visited hypertext area he is looking at. It may happen that he looses the orientation, and 'gets lost in hyperspace'. This article presents a software architecture, that provides a visitor of a limited, distributed internet area with additional structure information depending on his individual navigation path. The knowledge of the local hyper-media area is made available by a server program and passed on to a visualization tool working in a frame of the client's internet browser. This tool is loaded automatically from the server as a Java applet. By communication with the server, the visualization tool remains active throughout the navigation process and can be updated automatically.

1. Introduction

Among the most important programs which are necessary to realize the communication in the internet are internet servers like Apache and browser programs like Netscape, Hotjava or Internet Explorer on the client's site. It is the task of an internet server to answer requests coming from the internet for data objects it has access to. A browser program makes it possible for a user to start such a request and to visualize the received data. Recently such programs are able to interpret languages like Java or Java Script [4]. Thus a more sophisticated presentation of information, e.g. by animations is possible. Moreover the communication with the user and an internet server can be improved. Server and browser are connected by the internet. For security reasons so called proxy servers are often placed between them. They can also have the function to enable a faster server response by caching the data and especially relief the network.
The search for information in the internet is usually done in two different ways. First, there are so called search machines like e.g. Alta Vista or Yahoo that have analysed and indexed a considerable part of the internet. Here one can start requests with the help of search words.
The second possibility, i.e. the browsing or navigating in the hyper-space, is helpful if one has a rough idea where, i.e. near which URL, the wanted information is located. An inquiry at a global search machine is often too inexact and a search machine on the local server often is not offered or would demand the formulation of a precise inquiry. Browsing as a search method is especially suitable if such a precise inquiry is difficult to formulate, i.e. the client has only got an inconcrete interest. It may also be the aim to scan an internet site with respect to certain preferences (search scan). Browsing is especially suitable for multimedia data when a textual inquiry is hardly possible. The different browsing strategies where examined by [2].
The problem with browsing is that a user soon looses his orientation. Without knowledge of the hypertext structure he chooses from a great number of decision possibilities and forgets former ones. A sequence of different 'views' of a hyper-media database is presented, as defined in [5]. The 'bigger' the views are ('big' means the number of decision possibilities) and the longer the necessary navigation paths are, the more confusing the hypertext will appear in the navigation process.
There are different ways to build navigation support tools. In [6] and [1] methods are presented that show

the (estimated) relevance of links for a user. It is also possible to generate web pages that are adapted to a user [7]. Another approach to realize a navigation support is the presentation of link usages of former users as presented in [8].
It also seems to be desirable to create a view of the hypertext structure depending on the navigation path of a user. This could be used to visualize the former path of a client or to visualize the local surroundings in hyper-space.
This visualization task is dealt with in this article. It is about a technical implementation of such a system on the client's and the server's site. A new software architecture is introduced that makes it possible for the client to automatically load a visualization tool as a Java applet only once at the beginning of the navigation support. Communication with the server keeps it active during the navigation process. The structure information is presented in a second frame window of the internet browser, the first window presenting the original web page or data object. Thus a clear and detailed presentation is possible that can be updated automatically during the navigation process.
In section 2 some technical basics are described and in section 3 the structure of the developed system is presented. In section 4 the technical implementation and some possible visualization examples are discussed. The article ends with the conclusions and an outlook in section 5.

2. Technical Basics

Different technical solutions were presented or developed for the project that solve parts of the considered problem. They will be presented in the following section.

2.1. Methods to analyze distributed hyper-media structures

The first task that has to be considered is the extraction of knowledge about the hypertext structure. If only the link structure is considered, it is sufficient to examine the link connections of the data objects on the internet servers. The method that was used here is similar to the one presented in the MOMspider-project [3]. The well-known scan program 'wget' could also be used here.
The programs load the considered start web pages. These pages are parsed. The addresses of internet pages and other data objects are extracted. Then the scan process continues recursively at the (local) data objects where the found addresses point to.
The list of the scanned internet objects can easily be used to construct a graph structure of the considered internet area. The HTML pages or multi-medial objects can be regarded as knots and the hyper-links as edges of this graph.

2.2. Localization of hyper-media structure knowledge

One question that has to be considered is the localization of the knowledge about the hyper-media structure. It would be possible to store this knowledge on the client's site. This means that prior to the actual search (browsing) process an analysis is performed of the considered internet structure with the described methods. Since such an analysis might take a lot of time and since the client may visit a lot of different internet servers some of which he may visit spontaneously, this doesn't seem to be a good method.
In the system presented here, the knowledge about the hypertext structure will therefore be stored on one of the internet servers of the considered hypertext area or on a server in the near surroundings. The described system can be understood as a service that is offered by the provider of a limited number of internet domains.

2.3. Tracing of a navigation path

For the described problem it is necessary to registrate the navigation path of a certain user with a control server. A first idea for this tracing task is the analysis of the data stored by internet servers automatically when a client requests data. These files are called 'access-log' files. Since we consider a number of servers, different files would have to be considered. However it may happen that some data objects are stored by proxy servers as described above. Thus not all requests will reach the servers which makes it hardly possible to estimate the actual navigation path.
These problems can be solved by a method described by [1]. The navigation of the user starts by requesting a certain web page. This page can e.g. be the modified starting page of an internet area, in our case the starting page of a university. The page is modified so that all links point to a certain server, that is called 'trace server' in the following. The modified links contain as additional parameters the originally requested page address and some other parameters. The following address

$$http://server/page \tag{1}$$

can be modified to

$$http://proxyserver?server/page + id \tag{2}$$

where '*proxyserver*' is the web address of the trace server and the '+'-symbol separates the additional parameters.
One of the additional parameters must be an id number that has been given to a certain client to track his individual navigation path.
When a client uses a link on the modified page, the request will be sent to the trace server. The trace server can analyze the request parameters and order the originally requested page from the internet. The links on this page will be modified in the same way and the trace server sends it back to the client. This redirection method is shown in Fig. 1.
If the trace server registrates the requests, it is possible to track the navigation path of a certain (anonymous) client in the hyper-media structure.

2.4. Visualization on the client's site (Java applets and browser frames)

There are different possibilities to visualize the structure information. The most simple method would be to send a structure view as a picture or a visualizing Java Script at each navigation step to the client. This view could be presented on the actual html page or in a separate window using frames. However, the presentation on the web page leads to a layout problem. The space is limited, which makes a clear structure presentation hardly possible. We propose to present the structure views in a second browser frame window, or in an external window connected with a second browser frame window as enabled by Java [4].
A presentation in another frame leads to the problem of updating the structure view. At every navigation step a new frame window has to be created, the external window has to be positioned by the user. A direct communication between the structure frame and the hypertext page, i.e. between two applets in two different browser frames is not possible.
Therefore we propose the following method.
The browser window is subdivided into two frames. One frame window is for the optically unchanged presentation of the link-modified internet page (or object). The second frame contains a Java applet. The two frames, the web page frame and the applet frame (containing some buttons for the regularization of the structure presentation by the user), are shown on the right in Fig. 4. The applet controls the structure presentation in the external window, shown in the the left window in Fig. 4. The applet also organizes the loading of a new internet object. The necessity of this is described in the next section.

2.5. The problem of applet activity during the navigation process

When a user navigates in the hypertext space of a certain server, the presented internet objects shown in one frame and the applet in the other frame remains active. If the user takes a link pointing to another server, the frames will be deleted and replaced by the new data. The applet will become inactive.
It is the aim that the applet remains active throughout the whole navigation process in the considered internet area and to be able to present the data and to organize the communication with the server.
To achieve this, the applet has to control the loading of the internet page. This can be done by using the method in the AppletContext-class of Java

$$showDocument(url, string), \tag{3}$$

which enables the applet to show an internet page in the other frame window [4, pp.152] and keep the visualization of the structure. It is however necessary for the applet to communicate with the trace server. This will be examined together with the description of the system architecture in the next section.

3. Description of the system structure

In the following the structure of the system for the navigation dependent visualization of local hypertext spaces is described. Figure 1 shows the global structure. The communication of a client (browser) with a local, limited internet space is changed by redirecting the communication through a trace server, as described in section 2.3, shown by the connections 1 and 2. Thus the navigation path of a user can be registrated. Utilizing a software to analyze the internet space as described in section 2.1, the internet structure has been scanned (offline, link 3), and the structure knowledge has been stored in a structure knowledge database. The trace server communicates with this database and can send the structure information to the client (link 4) together with the modified web page shown by link 5 in Fig. 1.
In the following, the trace server, the program on the client's site and the communication that is necessary to achieve the navigation stability will be described.

3.1. Server program

When the user downloads a certain (start) web page, he receives a modified page as described in section 2.3. Every link request used on this page will go to the trace server at first. As shown in Fig. 2 the server begins

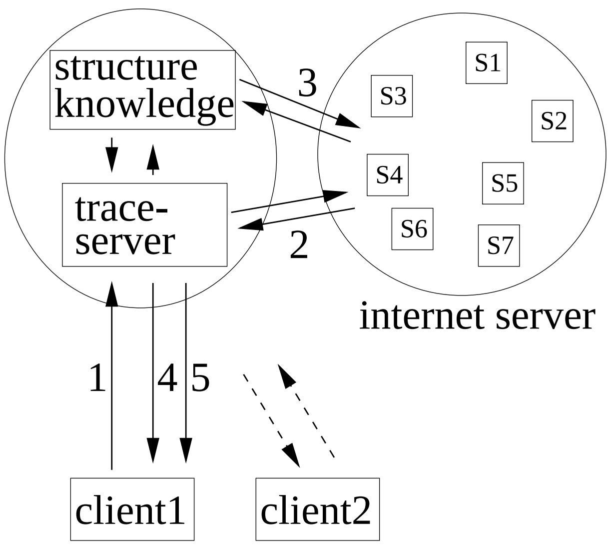

Figure 1. Structure of the navigation support tool; the trace server is located near the considered internet servers.

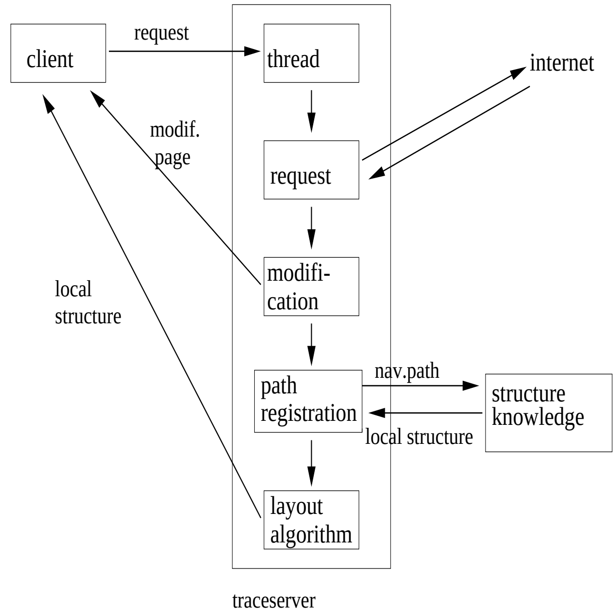

Figure 2. Sequence of the trace server processes

by starting a thread or process for this client. The thread downloads the requested page from the internet. It modifies the page i.e. the addresses on the page as described above and sends it back to the client. An identification number placed in the query string makes it possible to registrate or update the navigation path of the specific client. Now the structure knowledge database is requested to generate a local view of the database. After calling a layout algorithm the structure information is sent back to the client.

3.2. Client program

Fig. 3 shows the software structure on the client's site. As described in section 2.4 the browser window was subdivided into 2 frames. One frame contains the running applet program that was loaded at the beginning of the navigation support. The applet controls the structure presentation in the (external) structure window by communication with the trace server. It also controls the loading of an internet object in the second frame.

The modified internet page isn't sent to the client directly because it could replace the frame windows. Instead, an order is sent to the client. It tells the client applet which internet page the user requested and has to be shown in the other frame window. The applet requests this page actively with the method described in section 2.5 . Since the applet shows the internet object in the other frame, the frame window with the applet isn't replaced and the applet continues to be active.

4. Technical realization and visualization examples

The presented system was developed using the languages Java and C^{++} . The communication was realized with the help of socket and URL connections. It was tested with the Netscape Navigator 3.02. In this section some presentation examples are shown, which can be provided by the described software architecture. The most obvious possibility is the presentation of the navigation path of a user as shown in Fig. 4. Fig. 4 shows on the left a presentation of the navigation path in an external window while on the right in the right browser window the (optically unchanged) html page can be seen. In the left browser frame, where the applet is located, some buttons were added to control the presentation.

Such a presentation can help the user to orientate in hyper-space. In the same way as one can avoid getting lost in a maze by marking the way with a thread, the path visualization can avoid the loss of orientation in the hyper-space.

Another possible presentation is the visualization of the

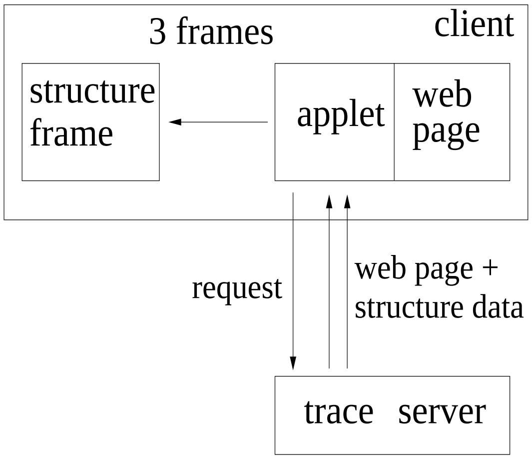

Figure 3. Software structure on the client's site, the browser window is subdivided into 2 frames, one containing a web page and the other containing the java applet. The applet controls the structure visualization in a third frame window and the loading of the web page.

decision tree of a user. Browsing through the hypertext a user makes decisions i.e. link usages at every navigation point. Former decision possibilities are no longer present, when a new 'view' is shown. In 5 the left window shows the decision tree of a user along his navigation path. If a new decision turns out to be not interesting, he can recall immediately the optically presented former possibilities.

5. Conclusion and future aspects

Browsing in hypertext as a method to look for information is the preferred method in many situations. Information visualization concerning hypertext structures can contribute a lot to keeping the orientation in a complicated structure of data objects and connections between them. During the browsing process it is necessary that the presentation of the structure is made dependent on the navigation path.

A method to implement such a system has been presented in this article. Methods were discussed concerning the extraction of structure information in a hypertext site. Moreover, methods were presented to make the structure data available to the user.

One decisive characteristic of the system is that the visualization tool on the client's site has to be loaded only once (automatically). By communication with the developed server program the visualization tool remains active throughout the navigation process. No plug-in or other program has to be installed on the client's site. Using the software it isn't necessary to modify the original internet data. All modifications needed to track the client's path are done online, using the original data objects. Two possibilities to make use of the system were presented, the navigation path and the decision tree visualization.

Many other similar views of the structure are possible e.g. a lookahead view where successive decision possibilities are shown. Future work could also deal with the analysis of prediction methods to provide a navigation support by estimating likely navigation steps.

As can be seen in Fig. 5, a hypertext structure view can be very extensive. Here the problem of a clear presentation of graphs becomes obvious. Knots shouldn't overlap and edges should be easy to see.

Another point that can be improved is the updating process of the structure knowledge. It would be much faster to scan the directories on the considered servers directly than to use the method described in section 2.1. The presented software was tested for Netscape Navigator 3.02. If other browsers are used or higher versions a further developed security system e.g. concerning the sockets must be considered.

References

[1] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Web watcher: A learning apprentice for the www. *AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, March 1995.

[2] E. Carmel, S. Crawford, and H. Chen. Browsing in hypertext: A cognitive study. *Transactions on System, Man and Cybernetics*, 22(5):865–883, Sep/Okt 1992.

[3] R. T. Fielding. Maintaining distributed hypertext infostructures: Welcome to momspider's web. *WWW94*, pages 25–27, May 1994.

[4] D. Flanagan. *Java Foundation Classes*. 1 edition, Sep 1999.

[5] G. W. Furnas. Effective view navigation. In S. Card, J. Mackinlay, and B. Shneiderman, editors, *Readings in Information Visualization, Using Vision to Think*, 1999.

[6] T. Joachims and D. Mladenic. Browsing-assistenten, tour guides und adaptive www-server. *KI*, (3):23–29, 1998.

[7] M. Perkowitz and O. Etzioni. Adaptive web sites: an ai challenge. *Internat. Joint Conference on AI (IJAI)*, 1997.

[8] A. Wexelblat. History-based tools for navigation. *Proceedings of the 32nd Hawaii Internat. Conf. on Systems Sciences (HICSS-32)*, 1999.

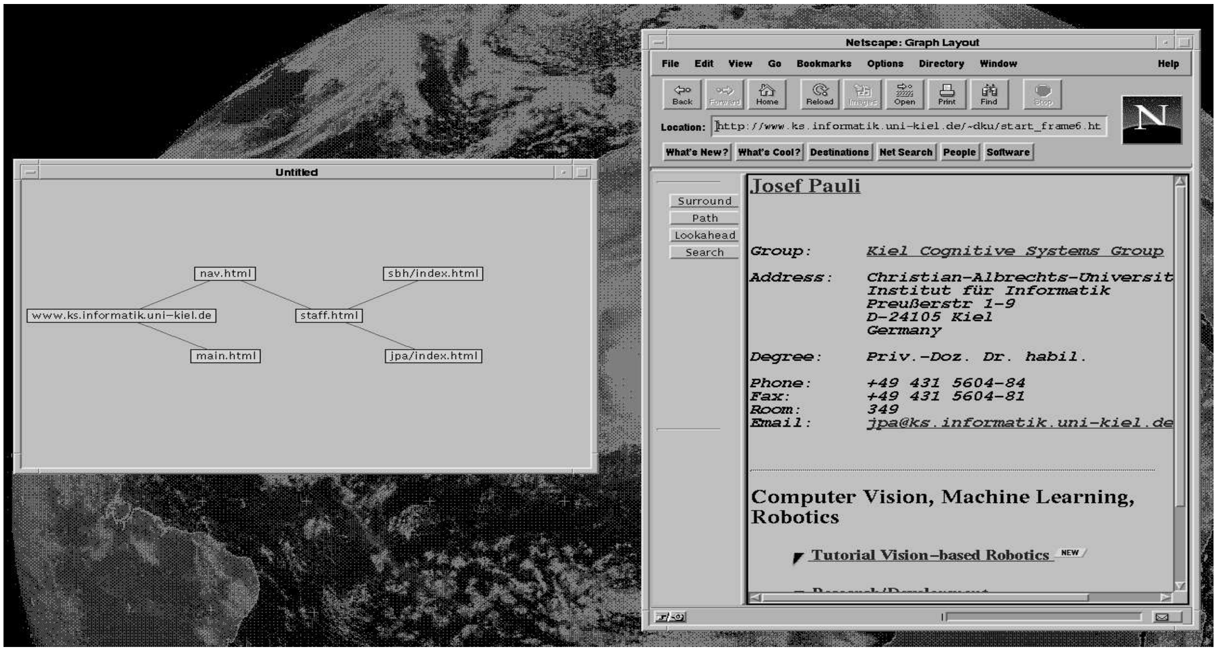

Figure 4. Presentation of the navigation path. The left frame shows the navigation path of a user, the right window shows the browser with two frames, the original web page on the right and the applet frame on the left.

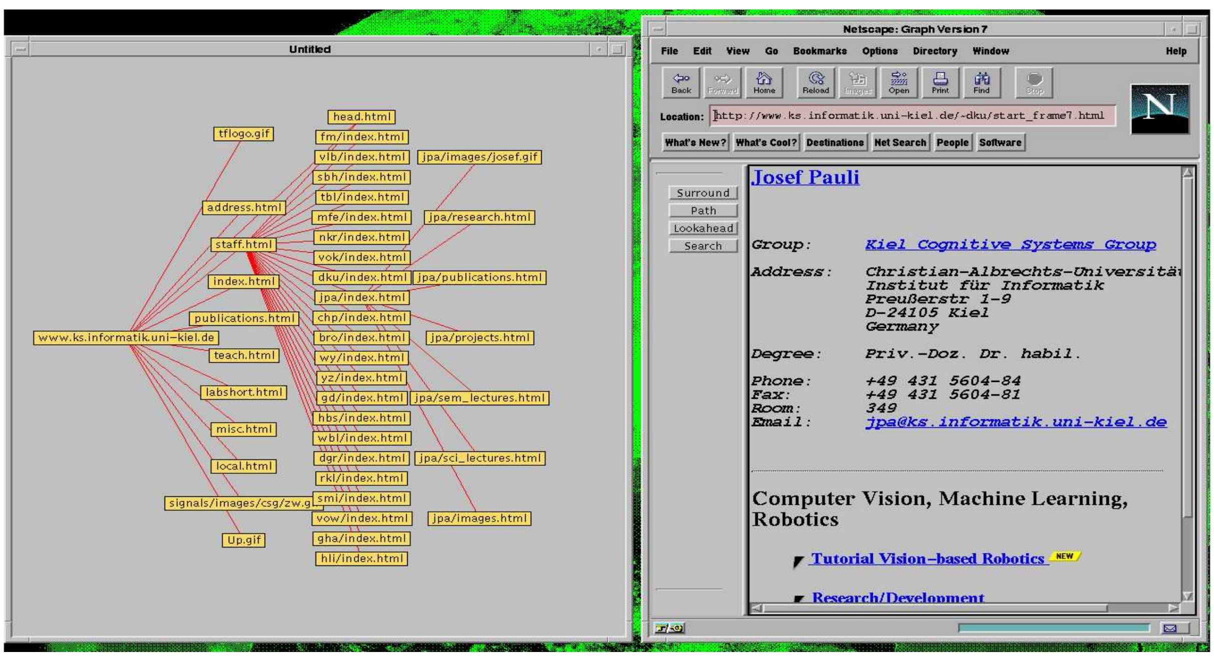

Figure 5. Local hypertext surroundings of the navigation path. The left frame shows the decision possibilities of a client along the navigation path.