

Visual preview for link traversal on the World Wide Web

Theodorich Kopetzky^{*,1}, Max Mühlhäuser¹

Telecooperation Department, Johannes Kepler University Linz, Altenbergerstrasse 69, 4040 Linz, Austria

Abstract

This paper demonstrates a technique for augmenting current World Wide Web browser implementations with features found in classical hypertext applications but unknown to the World Wide Web community until now. An example implementation is shown using Netscape Navigator 4.x using JavaScript, dynamic HTML and Java. The implementation follows an architecture based on a proxy server which acts as a gateway between the Internet and the browsing client. Based on the detailed example, support for further features is discussed. © 1999 Published by Elsevier Science B.V. All rights reserved.

Keywords: Hypertext navigation; User interface; Link preview; World Wide Web

1. Introduction

A novice user of the World Wide Web is usually impressed about the vastness of information available. At a second glance the impression gradually turns into desperation when the user tries to actually find valuable information. The usual solution is to contact a search engine like Altavista [1] or a Web portal like Yahoo [25] which usually work as a starting point for global navigation.

But even when navigating through small Webs (or hyper-documents), users tend to have navigation problems; in essence, these problems boil down to the decision about which links to follow and which to ignore. Considering a state of the art of about 12 years, little improvement was made. It was as early as 1987 that Conklin [9] mentioned the phenomenon in his late introductory article. Zellweger et al. [26] note that it is especially burdensome to follow a link

just to find out that it is not relevant for the reader. They state that the reader leaves the source context just to return to the previous reading position after determining the uselessness of the chosen link. We will briefly review what we consider three levels of support for link navigation.

(1) *Textual hints.* Landow [15] appealed to authors to enrich the context of link anchors in order to help users to decide if a link is worthwhile to follow. Furnas [11] discussed the advantages of links having a scent or residue of contents to come (this may be compared to “highway signage” where the sign shows, e.g., what is next and what is farther away). Zellweger et al. [26] introduce the notion of a *gloss*, which they define as a “brief explanation positioned in the margin or between the lines of a text”. A gloss may contain a description of the target of the link, meta-information and more. Some early hypertext systems like HyperTies [16] or the Guide system [5] already address some issues related to glosses. As to the World Wide Web, the Internet Explorer [17] supports a very basic kind of gloss by optionally

* Corresponding author.

¹ E-mail: {theo,max}@tk.uni-linz.ac.at

displaying a small pop-up window with a textual description of the link to be followed.

(2) *Typed links*. This concise and well-defined remedy to the link navigation problem described has been proposed early on, too. Halasz [13] required typed links, even composite links, in his late “seven issues” article, back in 1988. Manfred Thüring et al. [20] put up eight design principles and ten cognitive design issues to increase global and local coherence of hyperdocuments, one of them being “use typed link *labels*”. However, typed links are (i) not yet well supported, and (ii) not sufficient. Not well supported refers to the fact that HTML 4.0 offers support for link types like “contents”, “chapter”, “next”, etc. [23], but the standard browsers — like Internet Explorer [17] and Netscape Communicator [18] — currently do not support this feature yet. In the context of XML, a special-purpose XML linking language called Xlink [24] is emerging, but its widespread use is still far out. As to the attribute “insufficient”, the limits of typed links are obvious. (i) Unless a commonly accepted ontology is given for a field, the semantics of a given link type are known to the user but not to the reader. The latter must infer these semantics either from the type mnemonics (what might a link of type “in-depth” lead to?) from formal or informal descriptions (which distract the reader even more). (ii) Even if both author and reader have a common understanding of link types, the type information is just one kind of abstraction of the target contents. (iii) Additional authoring effort is required, so that typed links are neither a remedy for the bulk of existing World Wide Web documents nor can they be expected to be soon applied by the myriad of Web designers active in the Internet.

(3) *Further advancements*. Bieber et al. [4] categorize hypermedia environments in four generations in analogy to programming languages (machine languages, assembler languages, high-level programming languages, and finally packages like word processors which let users concentrate on what they want to do). They believe that current authors and readers of the World Wide Web are working in a second-generation hypermedia environment. Further, they discuss a set of hypermedia functionalities which they believe to improve the usability of the Web. One of these functionalities is “global and local

overviews”, to be achieved by overview diagrams and link previews. Obviously, textual hints and typed links as described above are just two facets of link navigation support. Among the more advanced concepts, work on information shape [10] is cited here just as a sample reference. This example illustrates a common problem with hypertext: many concepts have been studied in the past, but they have a long way to go until they are widely available for the World Wide Web community.

In this paper, we describe a technique to implement local overview for Web browsing, called “visual link preview”. The approach falls within category 3 described above (further advancements) and at the same time works with standard browsers without authoring effort; in other words, this approach can be made available to the World Wide Web community immediately.

2. Visual link preview

An author of Web documents has the following options to provide link information to the readers:

- He can describe in the context of the link by text or by image where the link will take the reader. This has been done for example in the image map in Fig. 1.

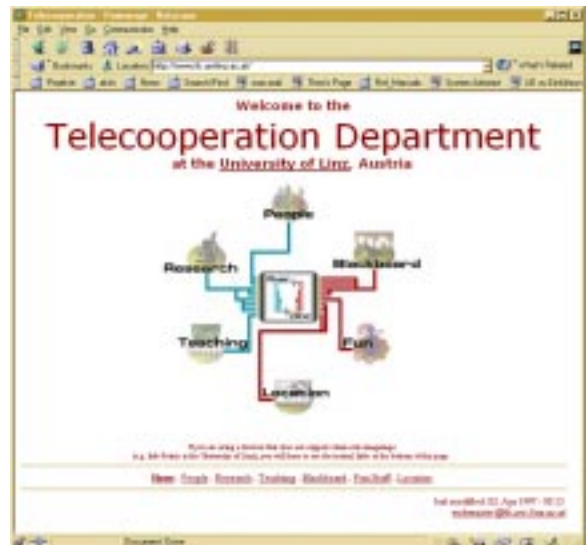


Fig. 1. The homepage of the Telecooperation Department.

- He can use the weak support provided by existing browsers. This support is limited to displaying a small text window.
- He can use dynamic HTML to enrich each link specification and may provide link information in another window or another frame.

All three possibilities demand work to be done by the author. But in the Web environment there is another source of information about the link: the target of the link itself.

Usually the reader does not get to see the link target until he has activated the link, leading to the problems stated in the introduction.

If the reader has already visited a region of the Web, he has built a mental map. Depending on the time passed the reader may need information in addition to the URL to remember that he has already visited this region. This is where the view of the page at the end of the link, provided as thumbnail preview, may help the reader's memory and will aid him in the decision of following the link.

Generating this thumbnail preview by hand can be done for small, static Web sites, but is unfeasible regarding large and fast changing sites on the Web. Thus automatic generation is necessary.

In addition to automatic generation these thumbnails have to meet the following requirements:

- (1) the image should be small to save bandwidth and server space;
- (2) the image should be in color; and
- (3) the quality of the image should be good enough to allow the reader to recall the accompanying image of the target of the link from memory; this is also the reason for point 2.

For the 1st and the 2nd requirement JPEG [14] was chosen as format for the thumbnail images because it provides small image files with a 24 bits color representation.

As for point 3, the quality of the preview images largely depends on the scaling algorithm used. Given the image in Fig. 1, two scaling algorithms are compared.

Below the difference between simple pixel resizing (faster) and bilinear resizing (slower, better quality) can be seen.

To improve recognition the slower method with better results was chosen. In addition, the second

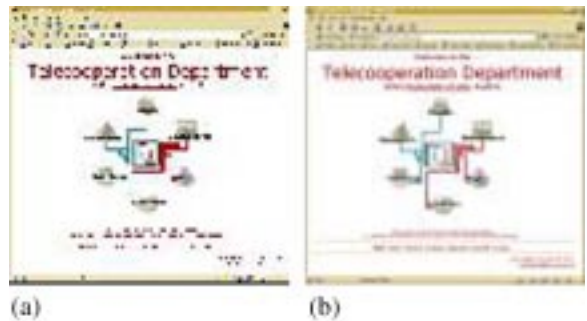


Fig. 2. (a) pixel resizing, 3628 bytes JPEG file. (b) bilinear resampling, 2884 bytes JPEG file.



Fig. 3. (a) Possible preview for a link to an anchor in a document. (b) Preview of a link to an E-mail address.

method usually yields smaller image files. To see the original image, take a look at Fig. 1.

2.1. Link types

Although explicit link types are part of HTML 4.0, no standard browser currently supports those.

But links also have implicit types, which can be used for preview purposes. The following list shows which link types are recognized by our system and how they are visualized. As the linking mechanism works with URLs we are using properties of the URLs to categorize a link.

- The URL points to the beginning of a Web page, as in <http://www.tk.uni-linz.ac.at/>. Links of this type are visualized using a thumbnail picture as in Fig. 2b.
- The URL points to an anchor, as in <http://www.encyclopedia.com/h.html#hypertext>². Links of this type may be visualized using a thumbnail picture or, if there is text after the anchor, the text referenced by the link itself may be displayed (see Fig. 3a).
- The URL starts with a protocol different from HTTP, like FTP, TELNET, MAILTO, etc. Al-

² This is a hypothetical link.



Fig. 4. Symbol for a dead link.

though these link types are rather self-explaining a symbol is used to represent the protocol used (see Fig. 3b for an example).

In addition to these properties links can have meta-properties, like *dead* or *forbidden*. If the reader knows in advance that a link is dead, he can safely ignore it. The symbol chosen for a dead link is presented in Fig. 4. These meta-properties are currently derived from the answer of the Web server where the link leads to, and so for example a 404-error is mapped to the dead end symbol.

2.2. Link preview

Fig. 5 shows how the preview looks like using the Netscape Communicator 4.5.

In case the reader requests a preview which is not available yet the reader is informed about that fact at the same location, as seen in Fig. 6.



Fig. 5. The mouse pointer (not shown) moved over the Blackboard symbol and the preview opened.



Fig. 6. No preview available information.

2.3. The presentation of the link preview

A document delivered by the proxy server does not look different to the reader than the original document. The link preview is simply activated by moving the mouse over a link.

The presentation of the link preview is animated. This means that the reader has time to accommodate to the new situation. The preview opens below the link and will remain open for seven seconds.

The preview can be closed by moving the mouse over the preview and then out of it. The preview will close also when the reader moves the mouse over another link and thus activating a new preview image.

To follow the previewed link, the user has the possibility to activate the link itself as usual or to click on the preview image.

3. Architecture

Our aim was to implement link preview without requiring support from the Web author. This approach has the advantage that all existing pages on the Web can be browsed without requiring any change to be made by hand. On the other hand the material has to be changed for the preview to work.

To solve this problem an approach using a proxy server was chosen. An overview of the main components of the proxy server, which has been implemented in Java, can be seen in Fig. 7.

The proxy server has the following tasks:

- analyze the links in a requested HTML document and generate the preview images for all links in the document;



Fig. 7. Basic architecture of the proxy server.

- cache the requested HTML documents and the computed link preview images for future access;
- modify the HTML documents in a way that the requesting browser is able to show the link preview images.

This approach has the following advantages:

- the proxy server has to generate the preview information only once (depending on server space);
- many readers can share one proxy server and thus benefit from already generated preview information;
- readers only have to make one change in their browsing environment: they have to configure the Web client to use a proxy server — everything else is done automatically;
- the proxy server can use other proxies servers and thus benefit from information already fetched from the Web.

3.1. Flow of events

A typical flow of events is described in the following steps (this assumes that in the beginning the required page is cached nowhere).

- (1) The reader enters an URL.
- (2) The browser asks the proxy server for the requested document.
- (3) The proxy server fetches the document from another proxy server or directly from the Web. After the document is fetched completely it is passed to the parser.
- (4) The parser analyses the document regarding its structure and searches for link information. If link information is found, for each link an internal Web browser will be started. These browsers will be used to generate the preview images. The document is modified to enable link preview by inserting the JavaScript code. After parsing is finished the modified document will be cached and passed to the requesting browser.
- (5) The requesting browser displays the document. Link preview images may or may not be ready yet.
- (6) If the reader moves the mouse over a link, the code inserted by the proxy server tries to load the preview image. If it is available at the proxy server, it will be transmitted and displayed, otherwise a notification regarding the missing image will be displayed (see Fig. 6).

Part of these activities happen in parallel to speed up processing.

4. Client-side implementation

To enable link preview at the client side, we had to utilize dynamic HTML. Unfortunately Internet Explorer and Netscape Communicator implement dynamic HTML each in their own way. The Internet Explorer uses cascading style sheets [22], which are not fully supported by Netscape's Communicator and the Communicator uses a technique called layers which is not supported by the Internet Explorer. We decided to implement the first version for the leading browser at that time (which was Netscape).

4.1. Layering

To display the preview information so-called layers are used. Layers can be viewed as small HTML documents which are displayed in the same browser window. Layers are displayed over the base window thus hiding part of it. In addition, layers may overlap other layers.

To enable the browser to show the link preview, the proxy server inserts a layer definition for invisible layers and a short JavaScript program at the beginning of each downloaded HTML-page. An example for a layer definition can be seen in Fig. 8. The position where the insertion actually takes place depends on the structure of the document (if it contains a head tag or a frame set, for example).

Additionally each `<A HREF>` tag is modified to react to events when the mouse pointer moves over the link. A mouseover event-handler is inserted into each link description. This handler will activate a procedure in the inserted JavaScript program if the reader moves his mouse over the link. The handler passes as a parameter the name of the preview image to the procedure. This name was inserted into the link definition just as the event-handler. The procedure will make the invisible layer visible and set some parameter of the layer, e.g. the information which preview image is to be shown.

Fig. 9 shows how a document looks like after modification by the proxy server. First there is the inserted layer and JavaScript code. Then follows the

```

<LAYER NAME= "preBase"
  LEFT="0" TOP="0" width="100" height="100"
  CLIP="0, 0, 100, 100"
  VISIBILITY="hide"
  onMouseOut="deactivate()"
  bgcolor=#FFFFCC>
</LAYER>

```

Fig. 8. A simple layer definition.

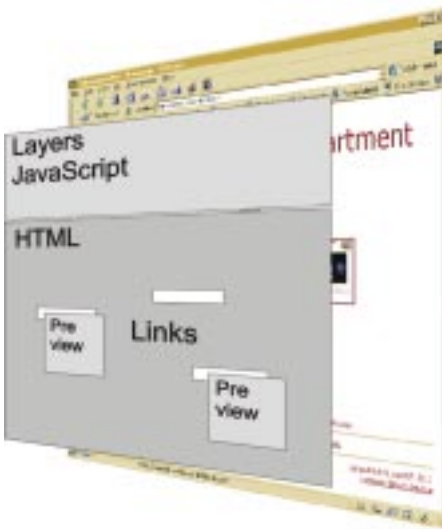


Fig. 9. Client side architecture.

modified HTML code which still may contain links without preview code (e.g. if the parser was not able to parse link information correctly). Preview images are fetched on demand from the proxy server.

5. Limitations

Unfortunately the current implementation has some limitations:

- It only works with the Netscape Communicator. If an Internet Explorer is configured to work with the preview proxy, no preview is available. However, normal usage is not restricted.
- The proxy server needs a lot of bandwidth to build the preview images because for each link encountered and not found in the cache a Web

browser is started. Imagine a reader following a link to a Web page containing a link index with hundreds of links. The proxy server will need some time to build all the preview images.

- The size of each HTML document transferred from the proxy server to the client increases approximately by 2 kb.
- It may interfere with already existing JavaScript and layer code in viewed documents. This can be averted by parsing the document for names used by our system and adapting naming conventions on the fly.

6. Related work

6.1. Classical hypertext

Different types of link preview or displaying the link type exists in several classical hypertext systems, like for example in Sepia [19] which displays the link label next to the link arrow in its link overview diagrams or in HyperTies [16], where a brief description is displayed at the bottom of the screen.

6.2. The Web

There exists an interesting concept which is called displets [21]. Displets provide a principled way to extend HTML. In their proof of concept implementation the authors implemented among others a displet for anchor groups (a way to implement multi-end-point links (see also [4]) and anchor labels. Because displets are implemented as Java classes, they may show any image as well but they lack the server-side component which generates the preview images.

6.3. Other user interfaces

A lot of different approaches exist to display an overview of large information spaces, like Pad++ [3], which uses an interactive panning and zooming navigation metaphor, or distortion-oriented viewing techniques like the generalized fisheye views [12], which employs a distorted representation of out-of-focus material.

Zooming has as disadvantage that the material in focus is moved out of focus if the reader zooms in on a detail. Another disadvantage is that zooming is difficult to integrate in standard Web browsers unless one is willing to implement a plug-in for a browser. Our approach integrates well and requires minimal user intervention to be set up.

Distorting techniques have the same limitations as mentioned with zooming when trying them to integrate in a browser and usually need their own window to work effectively. Our approach reuses the available screen space.

6.4. Link services

An interesting approach to enhance usability of the World Wide Web is the use of a link service [7]. Via link services readers can gain a surplus of information by applying different sets of links to one and the same Web document. Link services store link information external to the documents linked. Current implementations (as shown in [8]) do not store thumbnail information along with the link information. We think that by storing the preview image link services could gain attractiveness.

7. Future work

As already stated the current implementation only works for Netscape 4.x browsers. We are currently working on a version which will include full support for the Microsoft Internet Explorer as well.

In addition we are considering to implement the following features.

- Support for visualizing 1 to n links and selecting them in a screen estate saving way. An example can be seen in Fig. 10 which shows an excerpt

Wholly new forms of encyclopedias will appear, ready-made with a mesh of associative trails. Encyclopedia Britannica ; Electric library as at his touch the associated opinions and occasions of his whole experience, and of the experience of friends and authorities. The patent attorney has on call the millions of issued patents, with familiar trails to every point of his client's interest. The physician, puzzled by its patient's reactions, strikes the trail established in

Fig. 10. Excerpt from “As We May think”.

from “As We May Think” by Vannevar Bush [6]. A preview window opened below the link “wholly new forms of encyclopedias” revealing two links to choose from. This technique can be easily adapted to display more links.

- Displaying more information about a link. This could be, for example, the ping time to the server where the link is pointing to. Other interesting information could be keywords associated with a link (as long as provided in a format understood by our system).
- More control over the preview. Currently, control is limited to the states “preview on” and “preview off”. Users could, for example, define which link to preview in which manner.
- There are some interesting “fluid link” [26] techniques that we are interested in to mimic on the Web.
- As the proxy server approach is basically an intermediary approach as introduced in [2] we are looking into utilizing the power of the Web Browser Intelligence (WBI) architecture.

8. Summary

Visual preview of the target of a link can provide valuable clues for the reader when making the decision if to follow the link or not.

We have shown that a simple preview service can be implemented using standard Web browsing software and described an implementation which requires minimal user activities to be set up and does not need additional client software.

We are continuing to develop our approach adopting to new technologies as they become available and integrating novel navigating concepts.

Acknowledgements

I would like to thank Shai Kariv, Noam Ambar, and Amir Gur for letting me use the source code of their basic proxy server, and Ronald Angerer and Andreas Schwediauer for implementing parts of the system.

References

- [1] AltaVista, <http://www.altavista.com/>
- [2] R. Barrett and P.P. Maglio, Intermediaries: new places for producing and manipulation Web content, in: Proc. 7th Int. World Wide Web Conference, Comput. Networks and ISDN Systems 30 (1998) 509–518.
- [3] B.D. Bederson and J.D. Hollan, Pad++: a zooming graphical interface for exploring alternate interface physics, in: Proc. ACM UIST '94, 1994.
- [4] M. Bieber, F. Vitali, H. Ashman, V. Balasubramanian and H. Oinas-Kukkonen, Fourth generation hypermedia: some missing links for the World Wide Web, *International Journal of Human-Computer Studies* 47 (1997) 31–65.
- [5] P.J. Brown, Turning ideas into products: the Guide system, in: Proc. ACM Hypertext '87, 1987, pp. 33–40.
- [6] V. Bush, As we may think, *Atlantic Monthly* 176 (1945) 101–108, <http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm> or <http://www.ps.uni-sb.de/~duchier/pub/vbush/>
- [7] L.A. Carr, D. De Roure, W. Hall and G. Hill, The distributed link service: a tool for publishers, authors and readers, in: Proc. 4th Int. World Wide Web Conference, Boston, MA, USA, O'Reilly and Associates, December 1995.
- [8] L.A. Carr, D. De Roure, W. Hall, G. Hill, Implementing an open link service for the World Wide Web, *World Wide Web Journal* 1 (1998) 2.
- [9] J. Conklin, Hypertext: an introduction and survey, *IEEE Computer* (September 1987) 17–41.
- [10] A. Dillon, M. Vaughan, It's the journey and the destination — shape and the emergent property of genre in evaluating digital documents, *New Review of Hypermedia Multimedia* 3 (1997) 91–106.
- [11] G.W. Furnas, Effective view navigation, in: Proc. ACM CHI '97, *Human Factors in Computing Systems*, pp. 367–374.
- [12] G.W. Furnas, Generalized fisheye views, in: Proc. ACM CHI '86, 1986, pp. 367–374.
- [13] F.G. Halasz, Reflections on notecards: seven issues for the next generation of hypermedia systems, *Commun. ACM* 31(7) (1988) 836–852.
- [14] JPEG, Joint Photographic Experts Group, <http://www.jpeg.org>
- [15] G.P. Landow, Relationally encoded links and the rhetoric of hypertext, in: Proc. ACM Hypertext '87, 1987.
- [16] G. Marchionini, B. Shneiderman, Finding facts vs. browsing knowledge in hypertext systems, *IEEE Computer* (January 1988) 70–80.
- [17] Microsoft, Internet Explorer 4.01, <http://www.microsoft.com/windows/ie/ie40/>
- [18] Netscape, Netscape Communicator 4.x, <http://developer.netscape.com/>
- [19] N. Streitz, J. Haake, J. Hannemann, A. Lemke, W. Schuler, H. Schütt, M. Thüring, SEPIA: a cooperative hypermedia authoring environment, in: Proc. 4th ACM Conf. on Hypertext, Systems I, 1992, pp. 11–22.
- [20] M. Thüring, J.M. Haake, J. Hannemann, Hypermedia and cognition: designing for comprehension, *Communications of the ACM* 38(8) (1995).
- [21] F. Vitali, C.-M. Chiu, M. Bieber, Extending HTML in a principled way with displets, in: Proc. 6th WWW Int. Conference, *Computer Networks and ISDN Systems* 29(8–13) (1997) 1115–1128.
- [22] World Wide Web Consortium, Cascading style sheets, <http://www.w3.org/Style/>
- [23] World Wide Web Consortium, HTML 4.0 Links: document relationships: the LINK element, <http://www.w3.org/TR/REC-html40/struct/links.html#h-12.3>
- [24] World Wide Web Consortium, XML XLink Requirements 1.0, <http://www.w3.org/TR/NOTE-xlink-req/>, February 1999.
- [25] Yahoo, <http://www.yahoo.com/>
- [26] P.T. Zellweger, B.-W. Chang and J.D. Mackinlay, Fluid links for informed and incremental link transitions, in: Proc. ACM Hypertext '98, 1998, pp. 50–57.



Theodorich Kopetzky is a Ph.D. candidate at the Telecooperation department at the Johannes Kepler University of Linz. He received his master of Computer Sciences in 1993 and worked afterwards in the field of embedded systems. He began to work as a researcher at Telecooperation department in 1996. His research interests are hypermedia authoring, navigation, and web-based training.



Max Mühlhäuser is a full professor of computer science and head of the Telecooperation Research Group at Linz University, Austria. His research domain is software engineering for distributed multimedia systems, mobile and ubiquitous computing, and human-human and human-computer cooperation. He received his diploma and Ph.D. in computer science from Karlsruhe University, Germany, in 1981 and

1986, respectively. He is a member of ACM, the German Computer Society, and IEEE.