

The Look of the Link - Concepts for the User Interface of Extended Hyperlinks

Harald Weinreich, Hartmut Obendorf, Winfried Lamersdorf

University of Hamburg, Department of Informatics, VSYS
E-mail: {weinreich, 5obendor, lamersdorf}@informatik.uni-hamburg.de

ABSTRACT

The design of hypertext systems has been subject to intense research. Apparently, one topic was mostly neglected: how to visualize and interact with link markers.

This paper presents an overview of pragmatic historical approaches, and discusses problems evolving from sophisticated hypertext linking features. Blending the potential of an XLink-enhanced Web with old ideas and recent GUI techniques, a vision for browser link interfaces of the future is being developed. We hope to stimulate the development of a standard for hyperlink marker interfaces, which is easy-to-use, feasible for extended linking features, and more consistent than current approaches.

KEYWORDS: XLink, distributed hypertext, user interface, link marker, Web

INTRODUCTION

As part of a recent endeavor to extend the functionality of the Web, the World Wide Web Consortium (W3C) proposed XLink, XPointer and XPath [15; 19; 20] as linking standards for the Extended Markup Language (XML). While these specifications elaborately define the syntax of XML linking, visualization of and interaction with extended links remain largely undefined. Although current user perception of hypertext is mainly determined by Web experience and as a result links naturally seem to be blue and underlined, previous hypertext systems demonstrated other link visualization and interaction techniques. Some of these techniques do not only show the text in a more legible way, they will also be better applicable to complex anchors.

Historical approaches can help to find solutions for the Web of the future. A first look at link visualizations will help to become familiar with the associated problems. XML linking will bring fundamental changes in the way links work; therefore we try to examine the use of historical approaches for XML linking and attempt to find out where they fall short. Furthermore, we present a set of proposals for link visualization, thus hoping to fuel the discussion about the user interface of one of today's most important applications.

HISTORICAL APPROACHES TO LINK VISUALIZATION

Though hypertext systems frequently offer broad functionality, the essence of hypertext are machine-supported links that provide rapid access to modularized information nodes in non-sequential order [16; 38]. The visualization of link anchors by link markers and the interaction with links – the link user interface – are an important characteristic of usable hypertext systems. However, no standard link user interface exists and only few papers reflect on this topic.

When visualizing hyperlinks, two conflicting objectives must be balanced well: on the one hand the information presented has to be intelligible and the design should be as little restricted as possible, on the other hand links have to be marked and emphasized to be noticed and identified.

Links originating from graphics or parts of graphics are especially problematic. They can only be marked by changing the presentation of the original image. Historically, only few methods to mark graphics were explored, one reason being that in early systems the focus was usually put on text support. Although images were generally envisioned and often supported, embedded graphics did not have the dominant rank as link anchors as in the Web today. A well-known system that supported links from images was *Intermedia*. It marked anchors by an arrow icon above the image [46]. Mosaic's approach was hardly more mature: it drew blue boxes around all linked graphics. Serious conflicts with intended page design forced the majority of Web authors to hide the border. Instead, the hypermedia designer has to use other, *non-standardized* methods to make anchors visible. Often, 3D-elements, boxes, and rollovers are used to mark anchors. However, this can only work when the graphics are specifically created for the purpose of providing link anchors. Later changes in anchor placement would force to alter the graphics.

For textual anchors two objectives have to be balanced well: on the one hand the readability of the text has to be kept as good as possible, on the other hand the link anchors have to be distinguishable from other text. The appearance of text has been changed in a variety of ways to highlight the anchor region: typeface, style, size and color of the text have been altered, and graphical elements like arrows or boxes have been added.

Some systems, like Peter J. Brown's *Guide* [8], used different text styles like bold and italic types to mark anchors. This overloads the typographic significance of these attributes and limits the formatting options for unmarked text.

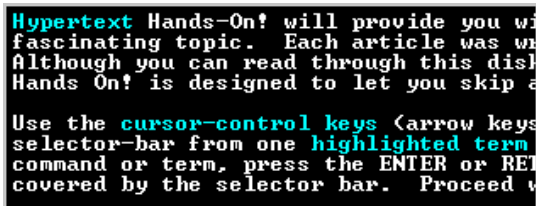


Figure 1: HyperTIES used cyan to highlight links [38]

HyperTIES (Fig. 1) avoided this problem by using a distinct text color for link markers, similar to Hyper-G's browser Harmony (Fig. 5) which utilized background colors [1; 38]. This has the advantage that the typeface and style of the text can be chosen freely.

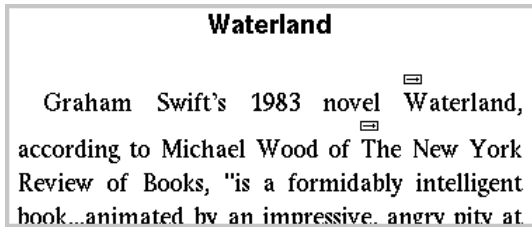


Figure 2: Intermedia's link marker arrows

IRIS' *Intermedia* marked hyperlinks with little arrow icons between lines of text (Fig. 2), showing the start of the link span but not its endpoint [46]. Emacs Info brackets link markers with asterisk symbols. These methods occupy extra screen space and change the layout of the text by inserting additional elements.

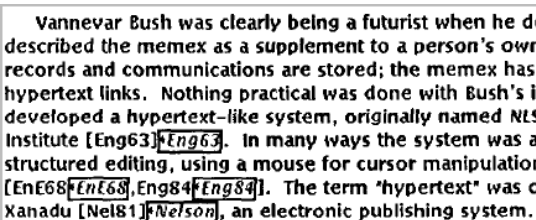


Figure 3: The Neptune Document Viewer [from 18]

Bernstein's Hypergate, the Neptune hypertext system (Fig. 3) and some early Web browsers UdiWWW (Fig. 4) drew boxes around the link marker text. This works with the layout but is quite obtrusive and distracting. An improvement to this technique could be found in HyperCard and Storyspace. They drew the boxes only when the reader pressed particular keys, making links evident on request thus keeping the text pristine the rest of the time.

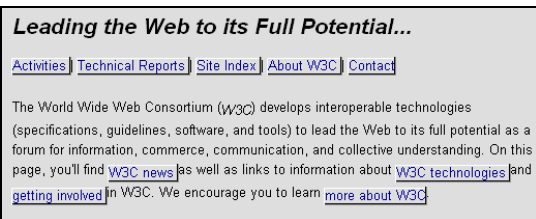


Figure 4: Anchor highlighting in UdiWWW

In fact, this was the consensus solution after the *Hypertext '87 demo sessions*, when hypertext designers could first compare all existing systems side by side [4]. Hiding links has many advantages: pages stay uncluttered, text stays readable, and page design is less influenced by link appearance. However, when using "links on demand" the interface designer must be aware of a potential disadvantage: since links are not always visible, possibly distracting mode switches have to be applied. Therefore this *link trigger* has to be seamlessly integrated into the interface.

A less desirable variant of this method was used by Symbolics Document Examiner [44]: link markers were hidden so well that they were only highlighted when the mouse passed over them. This forces a "hunt and peck" search for active regions. In Microcosm and DLS the user can query the system for invisible links by marking a word or some text and then issue a search for matching links [11]. Though anchors are not marked, this model is well applicable to generic links, given that most words can be selected as anchors.

Ignoring previous experiences, Mosaic returned to colored (blue) and underlined text to indicate link markers. This solution has the disadvantage to emphasize the link marker text permanently. The underlined markers stick out from the surrounding text and decrease its readability as underlining interferes with *descenders*, letters that drop below the line like p, q and j. The blue color is also an imperfect choice, since blue is hard to focus due to the chromatic aberration of the eye and furthermore especially elderly people have problems perceiving it. Growing old, the human eye gets progressively less sensitive to the color blue compared to other colors, making dark blue hard to distinguish from black [31]. The reasons for Mosaic's link marker appearance were obviously of technical nature: it was quite simple to implement, and at that time most computers had only 16 colors or a black and white display. Blue was the darkest of the available colors, closest to the black text; for monochrome displays, the text was underlined [35].

The pervasiveness of the Web has led us to accept underlined colored text as the de facto marker standard for all mainstream hypertext systems today. It can be found in help systems on various platforms and even in operating system components like the Microsoft *Windows Explorer*®. Though the development of user interface standards is desirable from a consistency point of view, user interfaces have otherwise improved, and we are still wedged with this historical hack.

More recent technologies like Cascading Style Sheets [6] allow to define the appearance of link markers in various ways. Also, the look of links can be configured somehow in current browsers, however, the *standard setting* is still blue and underlined, and links on demand are still not possible. Even worse, for the visualization of link maps in graphics no standard method can be found nor is implemented in Web browsers. This shows how important it is to design user interfaces well, considering earlier experiences, and

that even an interface with obvious design mistakes can become a standard.

ENTER XML LINKING

While the concept of links made it possible to create non-sequential texts, *rich* hypertext systems offer a more sophisticated functionality, i.e., they support comprehensive structuring, editing and navigation features [5; 46]. However, to date the Web itself only supports embedded one-way links. This limitation made the authoring of Web pages and the development of Web servers and browsers simple, enabling the Web to grow extremely fast. On the other hand, all approaches that try to integrate extended functionality into the Web have to utilize workarounds to overcome the weaknesses of this simple approach. With the advent of XML linking, the Web will be able to offer many of the features hypertext experts are missing today [43]. Standard browsers already migrate to XML and XSLT [14] and hopefully XML linking will become widely available soon.

The linking potential of XML linking¹ is based on two key standards which are necessary to create and describe links and link anchors:

XLink itself defines links as relation between resources or portions thereof. Syntactically, a link consists of an arbitrary number of resources (local or remote) and arcs. A resource is any addressable unit of information or service, while arcs create directed relations between two resources each [20].

XPointer allows to address different kinds of spans in XML documents [19]. They can vary from points to complex regions and can even be distributed over the document, e.g. a XPointer could be used to address the string "Goethe" in all citations in an XML file:

```
xpointer(string-range(//cite, "Goethe"))
```

Two types of semantic attributes are defined: machine-readable information is stored in the `role` and `arcrole` attributes and the corresponding human-readable textual information is kept in `title` attributes. This type information can be specified for the link as a whole, each endpoint of a link and for every arc.

XML linking also presents a solution for linking *into* other authors' read-only material, by addressing parts of the documents' structure. There is no need for tailored target anchors, which are embedded in the referenced document, any more. The importance of this can be seen from printed media, i.e., referring to distinct pages or paragraphs.

To summarize, XML linking will allow a multitude of new hyperlink features, among them:

- structure and contents may be separated;
- links may be bi-directional;

¹ XML linking may not only be used for hypertext links, but for any kind of application describing relations, associations or compositions of XML documents. However, this paper focuses on its use for hypertext.

- links may be typed;
- links may have multiple endpoints;
- anchors may be complex or overlap.

While the syntax of XLink has been elaborately defined, most presentation and behavioral aspects of links have been deliberately excluded from the model. Consequently, they could be moved into a style/formatting language similar to CSS for HTML. Still, a lack of ideas to enable the user to cope with this extended functionality exists. Nevertheless, some hints on presentation are inconsequently included in the standard. There is a vague notion of displaying title attributes to enrich link anchors semantically and computing role attributes to realize typed links [20]. Pop-up windows are suggested for links with multiple endpoints [7].

Technically, the chances seem to be good that XLink will succeed. The changes in browser capabilities since Mosaic suggest the acceptance of more extensive hypertext features: Forms, JavaScript, Java Applets and Flash animations are now widely used. Some extended link functionality is already being simulated with DHTML, showing pop-up menus with multiple destinations or extra information.

VISUALIZING EXTENDED HYPERTEXT FEATURES

The two well-known hypertext models Dexter [23] and HyTime [21] had features almost matching, and sometimes exceeding XLink, but no system ever fully implemented them [22, p. 42]. Nonetheless, many systems existed that were far ahead of their time and offered functionality that is not available in the Web today. This enables us to find ideas and detect problems by looking at all these systems and how they implemented and visualized hyperlinks. We map the user interfaces of these programs to the linking features that are made available with the introduction of XLink. We can thus discuss what is needed by the user to profit from the extended functionality.

Separation of Structure and Content

Several former Open Hypermedia Systems like Microcosm/DLS [11] or the Devise Hypermedia System [22] permitted to store links separately from documents in dedicated linkbases. Likewise, XLink will allow the separation of structure and content for the Web.

The external storage of links permits multiple linkbases to be used for single Web pages. These links may originate from the original author but also from other authors without write access to the original document, like a single user or members of a group².

The use of several different linkbases can result in an unintentionally *great number of links*³. Therefore, the user must be enabled to select the employed linkbases. An

² These additional links can be used to annotate and supplement the existing information with other information of personal importance.

³ These links may also *overlap* (see following sections).

example for such a method can be found in “Third Voice”, a browser plug-in that adds annotations to Web pages⁴ (Fig. 5). A part of its functionality is a service that adds links from an external linkbase to keywords. These annotation link markers are distinguished by orange underlines. Third Voice offers an extra tool bar in the head of the browser where the presentation of its links can easily be toggled.

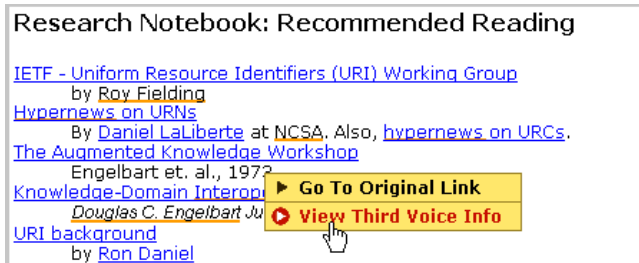


Figure 5: Third Voice adds additional links to an existing Web page and offers a choice of targets.

Microcosms/DLS already permitted the use of several linkbases. It offered a configuration screen to select the utilized link database from a given set [12]. Unfortunately this menu was not directly integrated into the browser interface and the addition of new linkbases was quite complicated. An XLink browser will also need the potential to find new linkbases⁵ and add them to a personal list. So far, no standard means has been established to do so.

Bi-directional Links

From the technical point of view bi-directional links help to keep links consistent and to avoid broken links. From the usability viewpoint they also permit to follow links backwards as opposed to the uni-directional “goto” links of the Web. A user could use this feature to find e.g. more recent information which is referring to an old but valuable document.

To benefit from bi-directional linking, the user interface has to support the backward traversal of links. Most hypertext systems with bi-directional links like Sepia, MacWeb or Hyper-G offered a *local map*, showing nodes and connecting links. This visualizes the topology and permits the user to select source objects directly on the map.

For the Web, the retrieval of links that refer to the current document poses a serious problem. A prototype Web browser tool described in [13] gathered this information from search engines. They alternatively proposed to extend the HTTP protocol to send backlink information gathered from the referrer URLs in the server log. The prototype offered a list of titles of Web documents that were linking to the current document.

⁴ Third Voice is available at <http://www.thirdvoice.com>

⁵ Furthermore, the primary linkbase will frequently change when browsing the Web, as it usually will be provided by the server hosting the current document.

Both approaches have their limitations if the number of links is high. Especially graphical maps use a lot of screen space if dozens of nodes and links are displayed. Thus, the number of objects has to be limited, e.g. by filtering the most appropriate ones.

Typed Links

A link type describes the relationship between source and destination of a link, often derived from semantic categories like “explanation” or “example” [41]. They were introduced to help users to get a better idea of a link target. Streitz et al. list semantic link information as their first principle of useful hypermedia system design [40]. However, typed links are only helpful if the user can distinguish the different types.

Tim Berners-Lee's WWW proposal [2] included typed links, and HTML allows Web authors to set the link type attributes `rel` and `rev`. Though, this feature is not supported by any current Web browser.

Sepia [40] and MacWeb [34] displayed the link type in an overview map close to the arrow visualizing the link. Once more, this link information is only available to the user if he considers two areas at the same time: a document view and a link map. He has to join these two information segments cognitively.

Other systems use text style to distinguish different link anchor types: the current Microsoft Windows help system displays explanatory pop-up links in green with a dotted underline and uses icons to indicate specific actions as the execution of a program. However, the potentials of Text style are quite limited, and inline icons can be distracting and create problems with the layout.



Figure 6: Different mouse pointers of the Guide system.

The Guide system utilized different mouse pointers to make link characteristics apparent [8]. The pointer changed according to the link type if it hovered over a link (Fig. 6). Since mouse pointers are independent from screen and text layout, this may be an interesting option for Web clients, too. Standard software, like word processors and graphics programs, and also operating systems, commonly employ these differently shaped mouse pointers as it is possible to indicate many different actions in a non-obtrusive, yet immediately visible manner.

Multiple Endpoints

Links with multiple endpoints do not connect only two, but a set of related nodes. Thus different alternative destinations can be provided. When a user initiates the traversal of a link with multiple endpoints, he can be requested to choose between the available options. This solution was preferred by most former hypertext systems. Microcosm and DLS

presented a list of generated link targets on an intermediary page as the result of a user query [11; 24]. Intermedia displayed a dialogue box with a list of link titles.

Likewise, the preferred idea for XLink seems to be a pop-up menu [20; 25]. Though lists of targets are probably the most straightforward approach, they may slow down Web navigation. A user has to make an additional selection from the pop-up list each time he follows a link.

Multiple links can also be used to automatically select the most decent destination by applying a filter. Already the father of hypertext Vannevar Bush suggested link filtering. If the user follows a *Guided Tour*, links of the displayed documents should be hidden [5; 9]. Intermedia could filter links by link attributes and Hyper-G by user rights. It would be even more desirable to filter by semantic criteria like a user's task or profile.

Complex Link Anchors

Many Web usability guidelines confine the setting and the length of link markers, e.g. Nielsen recommends that link markers should be about 5 words long [37]. This restriction is a concession to the limited link visualization potentials of current Web browsers, where extended link spans result in hardly readable underlined text regions. Hypertext systems that displayed links only on demand avoided these readability problems.

The XML linking standard allows arbitrary complex link anchors. As explained before, it is even possible to create discontinuous anchors, i.e., anchors that consist of several distinct regions. To the user this may appear like multiple anchors that share the same destination, which can be irritating. In Web system evaluations, already links that are displayed in more than one line have been found confusing, as the beginning and end of the anchor were not indicated by the browsers used [39].

Consequently, the extent of a link marker should be visualized. This is possible in recent Web browsers: the link marker can be highlighted if the mouse hovers over the link. However, the browser configuration has to be changed or an appropriate CSS must be defined.

Overlapping Link Markers

Link markers may overlap, either because an author creates two anchors at two intersecting text sections which are related to different destinations⁶, or because other authors create anchors overlapping with the link spans of the original author.

Hardly any current Web user will be familiar with the idea of overlapping link markers as they cannot be found on the Web or any popular hypertext system. Currently, it is not possible to create such constructs in HTML, since there is

⁶ Example: the phrase "psycholinguistics department" might be a link to the department home page, while another link explains the meaning of "psycholinguistics".

no way to distinguish different opening and closing anchor tags. This technological problem can easily be solved even with embedded links as Hyper-G's markup language HTF demonstrated. It used link identifiers to associate opening and closing link tags [33].

Nonetheless it is much harder to find a usable solution for the visualization of overlapping link spans. Harmony, Hyper-G's browser, used overlapping colored background boxes to mark the beginning and end of up to six overlapping markers (Fig. 7). But even two overlapping link boxes decrease the readability of the text and this method will finally fail if a larger number of anchors intersects: the boxes will shrink to pixel height, creating a very distracting background.

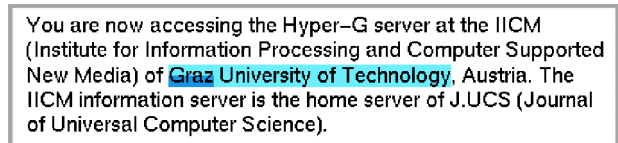


Figure 7: Link Overlap in Harmony.

The user must also be able to choose a desired link in the overlapping section. Third-Voice (Fig. 5) displays a pop-up window where the user can pick the link to follow. Harmony lets the user first *select* an overlapping link by single-left clicks and then *follow* it by a double-left click [1, p. 54]. Both solutions are not optimal, as the first one needs always two and the second one may even need an uncertain number of clicks to follow a link. The current version of Hyper-G does not support overlapping links any more⁷.

VISION FOR A BETTER HYPERTEXT USER INTERFACE

The Web, undoubtedly the most successful distributed hypertext system ever, has, despite its simplicity, serious usability problems. It must be prevented that this becomes worse when extended linking features are introduced.

We would like to revive a discussion by presenting ideas of an user interface strategy for extended links. To accomplish this we consolidated experiences of earlier hypertext research with established and innovative GUI techniques to create a consistent vision. These thoughts are widely based on the analogy of the hypertext reader as a traveler, introduced in Landow's authoritative "Rhetoric of Hypertext" paper. He divides the interaction with links into two key parts: departure and arrival [30].

Landow's ideas were based on his experiences with Intermedia. On the background of later hypertext research and the enriched linking capabilities of XLink a further discrimination is possible. The action of departure can be split into two sub-actions: first, the problem of locating the point of departure (identifying link markers) and second, the problem of getting sufficient information about the destination of the journey (understanding the link relationship).

⁷ HyperWave Information Server Version 5.5 uses HTML as markup language.



Figure 8: Mockup: Outgoing XLink anchors can overlap; they are highlighted using a translucent overlay. Note the marked scrollbar.

Considering the arrival procedure, the reader must get a reception at the destination to understand the extent and the context of the referenced material. The direction he came from, i.e., the origin of the journey, is the last page he visited and therefore known.

Finally, XLink not only allows for links that connect just two endpoints – it is also possible to build XLinks that represent whole paths or structures. Thus, XLink at last embodies a standard Web storage format for structural information, e.g. for guided paths or for hierarchical site maps. We will discuss the uses of these *hidden links* (hidden in the sense that they are not originating from rendered page content) in a separate section.

Point of Departure

Current methods of Web authors – emphasizing text anchors by using color and style and using specially tailored graphics to mark graphical link anchors – are already so common that they will probably continue to exist when XLink is introduced. However, as illustrated above, these methods do not have the potential to identify extended or externally defined XLinks. Furthermore, no prevalent standard visualization method can be found to identify graphical or image map links. Consequently, new schemata are needed to display supplementary links, e.g. from an external linkbase provided by an XLink service.

From the usability point of view, a consistent and uniform technique is desirable, that does not distract from reading and does not interfere with text and graphical layout techniques, but enables the user to identify even complex anchors clearly. We think that an appropriate way to accomplish this might be the use of *translucent* areas overlaid on the hypertext document. Overlays have the advantage to be feasible with text and graphics, indicating active areas directly by masking them. They can be applied

also at places where the document author did not plan a link. A possible distraction can be reduced by using soft and light colors for bright background and shady colors for dark background. User tests with more sophisticated translucent user interfaces showed promising results [17; 28].

An important factor that has to be considered is link density. If the ratio of marker area to unlinked area is high, the distinctive anchor appearance may overwhelm the “normal” text. Since an arbitrary number of “alien” links can relate to an XML Web page, a selection mechanism will have to prevent a phenomenon we would call *link overload*, similar to *information overload*, which could overshadow the interaction potentials of the approaching XLink Web. Therefore, the user interface must provide means to select which links will be put on view.

Once more, this calls for links-on-demand display techniques. The selection mechanism may be provided by an additional tool bar or window. A “link database browser” could be displayed at the left side of the window like the *history list* of Microsoft Internet Explorer or the *Sidebar* of Netscape 6. The tool would not only allow to select new link databases⁸, it would also permit to enable and disable linkbases, making them appear or disappear. Colors may be used to associate listed linkbases to the anchors on the screen (Fig. 8).

When a link starts from an anchor longer than a few words, the overall readability of the text decreases, at least as long

⁸ XLink offers a standard storage mechanism for external links. This permits the construction of *hyperbase systems* that offer compiled collections of links, e.g. as the result of a query [22, p. 167]. These services could be provided just like today’s search engines or Web catalogues.

as persistent highlighting is used. As with the introduction of XLink source anchors can become arbitrarily long, this question becomes increasingly important. We suggest a simple method to reduce the impact on readability: a narrow bar on the right side of the anchored paragraph. The use of different techniques for short and long anchors is suggested by looking at the use of conventional paper: markup on paper consists of highlighting words by coloring them with a translucent marker and, when longer passages need to be distinguished, marking whole paragraphs by using vertical lines on the page border. Sometimes a title is given to help recognize the underlying concepts of the passage. Markup of this style has the advantage of being apparent, but not as distracting as long, underlined text. It uses only little screen space and is goes along well with most layouts. Since this simple technique does not show the exact location of link marker start and end, it should be supplemented by a roll-over effect. The scrollbar, or optionally a small-scale overview window could be used to show the location of link anchors that are outside the currently visible page section. Using the scrollbar to locate particular areas on long Web pages was already suggested by [29].

Overlapping links of several linkbases could be visualized by translucent overlays in a defined neutral color, like bright gray. If the user clicks on such an area, a translucent pop-up appears, showing a list of the available link titles in the color of their associated database. Moving the mouse over these translucent items will highlight the related link markers in the document.

Destinations

At first sight the more or less uniform looking links of the Web are not typed. Apart from the marked text, the only preview a user can usually get in a Web browser is the destination URI [39]. Even this scant information is frequently utilized by Web users. Sometimes link titles or alternative descriptions to graphics are provided to hint at the content of the target document.

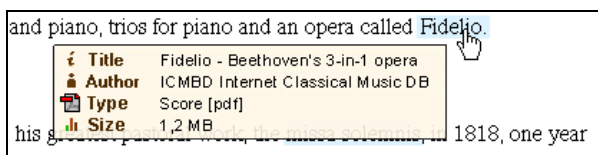


Figure 9: A pop-up menu that renders both XLink-specific, and other automatically gathered information.

Looking closer the current Web could already provide much richer information. Link targets can differ in type (“mailto:” links, downloads), availability (broken links), size, and connection speed (affecting download time). Further information of semantic nature like title, author and language of the target document or structural hints like indicating out-of-site links could be used to automatically enhance link preview. In our project *HyperScout* we already suggested techniques to display such information in pop-ups [45].

While XLink’s title information can also be straightforwardly displayed in such a pop-up window, the machine-

readable information is provided to compute type information. This can be used to induce alternative traversal behavior, or to get advance information about file types of target documents. It could also be used to filter links according to a specified user profile. If this leads to alternative browser behavior, this must not be hidden from the user. In addition to pop-ups, we suggest the use of different mouse pointers to immediately indicate link actions, comparable to the method of the Guide system.

Another promising approach are *fluid links*, introduced by Zellweger, Chang and Mackinlay [47]. The concept is called “fluid” as the document adjusts dynamically and makes space for a “gloss”, i.e., additional information to a link. This information appears between the lines of text or at the border of the page. The advantage of this technique is that the original document is not covered by a pop-up window, though a trade-off with increased distraction through the use of attention-intensive and time-consuming animation was observed.

If an XLink offers several destinations, the problem of selection occurs. Pop-up menus with a list of available links are suggested in the XLink definition and some publications [7; 20]. They have, however, the disadvantage to require additional user action: the user has first to choose a link, click and then he has to choose a target anchor and click again. We would suggest to use the role attributes to allow filtering, thus displaying only part of the link targets available. In certain cases it might even be desirable that a default destination is automatically selected when the left mouse button is clicked. Indicated by the mouse pointer, a pop-up appears only on right mouse click, presenting a choice of complementary link targets.

Arrival

The rhetoric of arrival in the sense of Landow requires that the reader gets the feeling of welcome at the destination document: “One must employ devices that enforce hypertext capacity to establish intellectual relations.” [30].

Establishing such an intellectual relation requires the user to determine the target of a link and its context. The method of today’s Web browsers to present the target is simple: the whole document is shown, or, if a fragment identifier was specified, the browser *tries* to scroll to the position of the fragment anchor. In fact this is a known usability problem of the Web: as the span of the link target is not visualized the user cannot identify the extent of the destination. If the fragment locator is near the end of a page, the browser often cannot scroll sufficiently down to display the link target at the top of the window.

Tim Berners-Lee’s first graphical Web browser on the NextStep system and later versions of Mosaic did already highlight the target anchor [10], a feature lost in current mainstream browsers. Since we already suggested translucent highlighting areas to identify starting anchors, a different method would be advisable to prevent misconceptions.

We again suggest a technique long before known to work on paper: lateral marker lines. A narrow bar on the far *left* side of the window is used to indicate the target span (Fig. 10). The chance of confusing incoming and outgoing links is thus (incoming links: left vs. outgoing links: right) kept to a minimum. The Devise Hypertext System utilized a similar technique to indicate target anchors [22, p. 314]. When a more precise visualization of the target becomes necessary (e.g. for tables), an on-demand method may be used: moving the mouse over the marker bar will shade the rest of the document except for the target area; a method already used in Harmony's PostScript viewer [1].

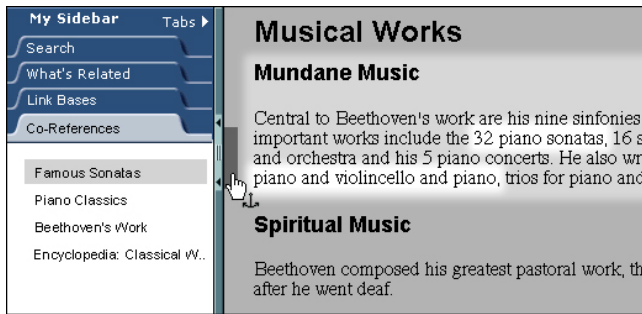


Figure 10: A mock-up showing the incoming link target in its context and information about co-references.

If the target section is larger than the visible window, clicking on the bar will “pin” the shading and the user may scroll the page. Additionally, the scrollbar may be used to show the extent of the target span compared to the whole page, especially useful if it does not fit into the window.

Although a very precise notion of the target anchor can be specified with XML linking, a weakness of XLink emerges: what the standard lacks, is a definition of the *link context*. Nanard and Nanard argue for a distinction of link anchor (as trigger) and link context (as minimum reading context) at both ends of the link. The link anchor is usually quite short and focused, as the link context is embedding the anchor (Fig. 10), enabling the reader to understand the relationship of a link better [27; 34]. The meaning of a sequence of words can change severely when torn out of its context, i.e., the surrounding sentence, paragraph or chapter. XLink misses an explicit definition of such context spans. This is a serious disadvantage, that could be easily fixed by an additional attribute for the resource tags. Then, if an anchor is selected, the context should be made visible.

It is also sometimes useful to supplement other links pointing to the target anchor, called co-references. These links could have been collected in earlier sessions, retrieved from search engines or compiled by *linkbase systems*. They can stem from material that was not visited in the course of the current search, and, if followed in the reverse direction, they can provide material related to the current target anchor.

We suggest to make the most appropriate co-references of the last navigated link available by right clicking on the lateral marker bar, just as the right click opens a pop-up on link anchors. A double click can be used to open a larger list

in the left side of the browser with more references pointing to that document. Finally, we think it would be feasible to apply filtering mechanisms utilizing the arc roles, e.g. to display only links that use the current anchor as an “example”.

The Use of Hidden Links

So far we have tried to optimize the visualization of hyperlinks with markers in documents. However, XLink can also be used to describe relations without markers, e.g. non-associative structural XLinks. These links can either be supplied by the author of a site or by an external source, e.g. a guided tour or a trail [9; 26]. Automatically generated link overview maps (local map, fisheye view, 3D landscapes) often seem to be more confusing than helpful, when used in large hyperspaces [42]. Because of the immense size and the distribution of the Web, structural information has to be provided for an overview that truly can help to find semantically related content.

A special link type was introduced in HTML 2.0 (<LINK REL>) to distinguish between structural and associative links [3]. Though this made it possible to separate structure-related and content-related navigation, it is poorly supported by current Web browsers. Only some less widely spread browsers like Lynx and iCab (Fig. 11) support the use of structural links. Thus, so far there are only a handful of Web sites that offer structural links. Yet, this information could often be easily provided, especially for generated Web content or sites created with an authoring tool.



Figure 11: iCab's structural link navigation toolbar.

To support structural information in XLinks special link roles and arc roles would have to be defined. This, however, would use the role attributes not for semantic but rather for syntactical information. Then again, it would be possible to provide complete structures, e.g. Guided Tours or Site Maps, in a single link, something not possible with the LINK element.

As for link markers, a consistent interface is needed for structural navigation: we suggest that XLink-aware browsers should provide an iCab-like toolbar for basic structural navigation. Furthermore a *hierarchical view*, like Hyper-G's “collection tree browser”, can be provided on demand. However, this additional navigation tool should be displayed in the same browser window, e.g. in place of the *sidebar*. The interface should also provide a standard interface for Guided Tours or other meta-structures, thereby eliminating the need for workarounds.

We can also imagine hybrid XLinks which bear structure, and have link markers in the Web page⁹. This implicit structure could be extracted and displayed in the standardized

⁹ Such structural links include: links on a homepage pointing into the site, site logos pointing to the homepage, arrows for next and last pages, etc.

user interface. The original embedded links should not be hidden: the user can thus either use the consistent standard interface (without having to search for navigation elements) or follow the rendered structural links (without having to leave the page context).

CONCLUSION

Usability has become a key factor for the success of software. Despite the intensive research on hypertext systems, no standards for hyperlink user interfaces have been agreed on. We are thus bound to the de-facto standard of the Web, a design with many inherent weaknesses that does not agree with extended linking features.

Experiences from software engineering have shown how to do better: the initial design of a system has to include its user interface as well as its functionality [36]. The representation of data is only of secondary importance.

Do we thus have to reconsider some current standardization processes? The XLink standard does hardly mention the user interface. The same lack of consideration of user interfaces is apparent in other W3C activities: Neither HTML nor other standards like SMIL or the Semantic Web Initiative mention design issues regarding the user interface. If these standards are not accompanied by concepts for presentation and interaction, an uncontrolled development similar to the HTML ``-Tag is to be feared. If moreover, inconsequently *some* interface issues are considered in the standard, this inconsistent approach can lead into usability disasters, since the novelty and complexity of extended XLinks may be misunderstood.

In this paper we try to stimulate a discussion on the visualization of and the interaction with extended hyperlink features¹⁰. We believe that this is necessary to prevent an impairment of Web usability when new linking features are introduced.

Experiences from historical systems can help to avoid mistakes and to provide solutions that are still topical. This paper presents problems and solutions for the presentation of and the interaction with extended hyperlink features. Though we are aware that the developed vision can still be enhanced, we gathered well-tried methods to create a consistent and easy-to-use interface model. In this process, design issues for XLink arose: we found some open issues, i.e., the missing definition of contexts, default arcs, syntax attributes or attributes needed to carry preview information (like the size of a target document). Some issues were completely left out, like the distribution of links via linkbases or an exact specification for the use of the semantic attributes.

Nonetheless, XLink can already be used today: when XLinks are employed on Web servers, the centralized storage makes link management much easier [32]. Using

XSL Transformations, XML (or XHTML) documents and XLink linkbases can be converted to HTML and be accessed by conventional browsers. In the long run, however, this functionality should be moved to the client – only then the browser will be able to exploit the full power of XLink. The success of XLink or a similar standard will eventually depend on two factors: decent tools for authors and readers.

REFERENCES

1. Andrews, K.: Browsing, Building, and Beholding Cyberspace, New Approaches to the Navigation, Construction, and Visualisation of Hypermedia on the Internet, Ph.D. Thesis, Graz University of Technology, 1996, <ftp://ftp.iicm.tu-graz.ac.at/pub/keith/phd/>
2. Berners-Lee, T.: Information Management: A Proposal. CERN Internal Communication, Graz, 1989, <http://www.w3.org/History/1989/proposal.html>
3. Berners-Lee, T. and Conolly, D.: Hypertext Markup Language Version 2.0, MIT / W3C, Boston / Graz, http://www.w3.org/MarkUp/html-spec/html-spec_toc.html
4. Bernstein, M.: HypertextNow: Showing Links, Eastgate Systems Inc., 1996, <http://www.eastgate.com/HypertextNow/archives/ShowingLinks.html>
5. Bieber, M., Vitali, F., Ashman, H., Balasubramaniam, V. and Oinas-Kukkonen, H.: Forth Generation Hypermedia: Some Missing Links for the World Wide Web. International Journal of Human Computer Studies, Vol. 47 (1), Academic Press, 1997, pp. 31-65
6. Bos, B., Lie, H. W., Lilley, C. and Jacobs, I. (Eds.): Cascading Style Sheets Level 2 Specification. World Wide Web Consortium, 1998, <http://www.w3.org/TR/REC-CSS2/>
7. Bosak, J. and Bray, T.: XML and the Second-Generation Web, Scientific American, May 1999
8. Brown, P. J.: Turning Ideas into Products: The Guide System, Hypertext'87 Proc., ACM Baltimore, 1987, pp. 34-45
9. Bush, V.: As We May Think, The Atlantic Monthly, July 1945, reprinted in: Interactions, Vol. III.2, 1996, pp. 35ff
10. Caillau, R. and Ashman, H.: Hypertext in the Web – A History, ACM Computing Surveys, Vol. 31 (4), December 1999
11. Carr, L., De Roure, D., Hall, W. and Hill, G.: The Distributed Link Service: A Tool For Publishers, Authors And Readers, Proc. 4th WWW Conference, Boston, Massachusetts, USA, 1995
12. Carr, L., De Roure, D., Hill, G. and Hall, W.: Web Links as User Artefacts, Techreport, University of Southampton, UK, 1996
13. Chakrabarti, S., Gibson, D. A. and McCurley, K. S.: Surfing the Web Backwards, In: Proc. of WWW 8 Conference, Toronto, Canada, 1999
14. Clark, J. (ed.): XSL Transformations (XSLT) Version 1.0, World Wide Web Consortium Recommendation, Nov. 1999, <http://www.w3.org/TR/xslt>

¹⁰ We created a repository for example visualizations at: <http://www.hyperscout.org/lookoflink/>

15. Clark, J. and DeRose, S. (eds.): XML Path Language (XPath) Version 1.0, WWW Consortium Rec., November 1999, <http://www.w3.org/TR/xpath>
16. Conklin, J.: Hypertext: An Introduction and Survey. IEEE Computer 20 (9), September 1987, pp. 17-40
17. Cox, D. A., Chugh, J., Gutwin, C. and Greenberg, S.: The Usability of Transparent Overview Layers. Summary Proc. of CHI'98 Conference, ACM Press, 1998, pp. 301-302
18. Delisle, N., Schwartz, M.: Neptune: A Hypertext System for CAD Applications, Proc. of Conference on Management of Data, Washington, USA, 1986, pp. 132-143
19. DeRose, S., Maler, E. and Daniel, R. (eds.): XML Pointer Language (XPointer) Version 1.0, World Wide Web Consortium Reenter Last Call, January 2001, <http://www.w3.org/TR/xptr>
20. DeRose, S., Maler, E. and Orchard, D. (eds.): XML Linking Language (XLink), World Wide Web Consortium Proposed Recommendation, December 2000, <http://www.w3.org/TR/xlink>
21. DeRose, S. and Durand, D.: Making Hypermedia Work: A User's Guide to HyTime, Kluwer Academic Publishers, Norwell, MA, USA, 1994
22. Grønbaek, K. and Trigg, R. H.: From Web to Workplace, MIT Press, Massachusetts, USA, 1999
23. Halasz, F. G., and Schwartz, M. D.: The Dexter Hypertext Reference, In: Proc. of Hypertext Standardization Workshop, Gaithersburg, USA, 1990, pp. 95-133
24. Hall, W., Davis, H. and Hutchings, G.: Rethinking Hypermedia, The Microcosm Approach. Kluwer Academic, NL, 1996
25. Halsey, B. and Anderson, K. M.: XLink and open hypermedia systems: a preliminary investigation, In: Proc. of ACM Hypertext Conference, San Antonio, USA, 2000, pp. 212 – 213
26. Hammond, N. and Allinson, L.: The Travel Metaphor as Design Principle and Training Aid for Navigating around Complex Systems. In: Proc. of the 3rd Conference of the British Computer Society HCI Specialist Group, 1987, pp. 75-90
27. Hardman, L., Bulterman, D. and Rossum, G.: Adding Time and Context to the Dexter Model, Communications of the ACM, Vol. 37 (2), Feb. 1994
28. Harrison, B. L., Kurtenbach, G. and Vicente, K. J.: An Experimental Evaluation of Transparent User Interface Tools and Information Content, In: Proc. of User Interface Software and Technologies (UIST'95), Pittsburgh, USA, 1995
29. Laakso, S. A., Laakso, K.-P. and Saura, A. J.: Improved Scroll Bars, CHI 2000 Extended Proc.: Interactive Posters, Den Haag, 2000
30. Landow, G. P.: Relationally Encoded Links and the Rhetoric of Hypertext. In: Hypertext'87 Proc., ACM, Baltimore, 1987, pp. 331-343
31. Lythgoe, J. N.. The Ecology of Vision, Oxford University Press, Oxford, 1979
32. Markos, J.: Out of Line Linking with X2X - Satisfying the need for better access to information, White Paper, Empolis UK Limited, http://www.empolis.co.uk/technology/tech_lin.asp
33. Maurer, H.: Hyper-G now Hyperwave: The next generation Web solution. Addison-Wesley, Harlow, UK, 1996
34. Nanard, J. and Nanard, M.: Should Anchors be Typed Too? An Experiment with MacWeb. Proc. of ACM Hypertext '93 Conference, 1993, pp. 51-62
35. NCSA: Mosaic(tm) for MS Windows User Guide, National Center for Supercomputing Applications (NCSA), University of Illinois, 1996, <http://www.ncsa.uiuc.edu/SDG/Software/mosaic-w/eleasinfo/2.1/index.html>
36. Nielsen, J.: Usability Engineering, Academic Press San Diego, USA, 1993
37. Nielsen, J.: Designing Web Usability: The Practice of Simplicity, New Riders Publishing, 2000
38. Shneiderman, B. and Kearsley, G.: Hypertext Hands-On! An Introduction to a New Way of Organizing and Accessing Information. Addison-Wesley, Reading, MA, 1989
39. Spool, J., Scanlon, T., Schroeder, W., Snyder, C. and DeAngelo, T.: Web Site Usability: A Designer's Guide. Morgan Kaufmann Publishers, 1999
40. Streitz, N., Haake, J., Hannemann, J., Lemke, A., Schuler, W., Schütt, H., Thüring, M.: SEPIA: A Cooperative Hypermedia Authoring Environment, In: 4th ACM Hypertext Conference Proc. (ECHT'92), Milan, Italy, 1992, pp. 11-22
41. Trigg, R.: A Network-Based Approach to Text Handling for the Online Scientific Community, Ph.D. Thesis, Dept. of Computer Science, Univ. of Maryland, Nov. 1983.
42. Utting, K. and Yankelovich, N.: Context and Orientation in Hypermedia Networks, ACM Transactions on Office Information Systems (TOIS), Vol. 7, No. 1, January 1989
43. Vitali, F. and Bieber, M.: Hypermedia on the Web: What Will It Take? ACM Computing Surveys 31(4), Dec. 1999
44. Walker, J. H.: Document Examiner: Delivery Interfaces for Hypertext Documents, Proc. of ACM Hypertext '87 Conference, Chapel Hill, USA, 1987, pp. 307-323
45. Weinreich, H.: Concepts for improved visualization of Web link attributes, Computer Networks, Vol. 33, 2000, pp. 403-416
46. Yankelovich, N., Haan, J. B., Meyrowitz, N. K. and Drucker, S. M.: Intermedia: The Concept and the Construction of a Seamless Information Environment, IEEE Computer, Vol. 21 (1), 1988, pp. 81-96
47. Zellweger, P., Chang, B.-W., Mackinlay, J.: Fluid Links for Informed and Incremental Link Transitions, In: Hypertext'98 Proc. (Pittsburg), 1998, pp. 50-57