inlineLink: Inline Expansion Link Methods in Hypertext Browsing

Motoki Miura, Buntarou Shizuki and Jiro Tanaka
Institute of Information Sciences and Electronics
University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
{miuramo, shizuki, jiro}@iplab.is.tsukuba.ac.jp

Abstract: Conventional Web browsers are designed to display a Web page for each window. A reader who wants to follow link with remaining currently shown webpage needs to choose “open link in new window” menu. In addition to the operation, the reader may control the position and dimension of the “new window” in order to appear both of the webpages at the same time.

To reduce the readers effort, we propose new browsing technique inlineLink, which shows linked web document nearby the link. inlineLink enables readers to follow the link efficiently. Because the readers become free from window management and back button. The reader can keep context of the current webpages and keep on reading task by using regular scrolling methods.

Keywords: hypertext, browsing technique, WWW, usability, Javascript, DOM

1 Introduction

“Following link” is a significant and intensively used method in browsing hypertexts. By following links, we can easily acquire information related to the current pages. The advent of HTML and graphical WWW browsers such as Nexus[1] and NCSA Mosaic[2] made the browsing so attractive.

1.1 Link following problems with conventional web browsers

While the concept of “link” is very simple and powerful, we often feel it difficult to read hypertext documents when we read many pages with conventional web browsers. The followings are the operations that we use to follow links in the web browsers.

Operation 1 Normal clicking on an anchor.

Operation 2 Open a menu on an anchor and choose “Open link in new window.”

Operation 3 Drag an anchor and drop into a window.

“Operation 1” is the most popular way to open a linked document. This operation is simple, intuitive and convenient. Therefore almost all pages are acquired by this method. This method works well unless the reader wants to go back to the previous document. “Back” button just allows readers to look at the previous page. When they press the back button, it is necessary to take their eyes off the document. Thus, the reading task will be interrupted. At the same time, the reader’s context of the document will be lost.
“Operation 2” is used in case the reader needs to show the linked document and previous one at the same time. This operation duplicates the current window. After that the view of the duplicated window turns into the linked document. The problem with this operation is that the duplicated window tends to overlap on the current window. In such case the reader must manipulate the size and location of these windows to show whole area on the display. These window management operations disturb the user’s reading tasks.

“Operation 3” may be useful if some windows are already opened and arranged on the screen without overlapping. The readers can display an interesting link to the any window they like. Moreover, window arrangement tasks such as resizing and layouting does not occur. But the reader should keep the relationships between the anchors and the windows in mind while reading.

1.2 Requirements for effective reading
We consider that the above problems come from unnecessary eye movement. Especially the situation that the reader needs to move a mouse cursor influences the performance of reading and comprehension tasks. Because the operation with mouse cannot be confirmed without any glance at the target objects.

Link following operations in conventional web browsers have not been designed to reduce the unexpected eye movement. As a result, the people could not concentrate on their reading lines on hypertext. Because the “following link” operation is often used and occupies the considerable part of the activity in hypertext browsing, reducing the amount of eye movement is required.

2 Concept of “inlineLink”
We have designed new link following technique “inlineLink,” which can reduce the distance of eye movement. An inlineLink method shows the linked document near the anchor by insertion. The text following is moved after the end of the inserted linked document. Figure 1 shows an ordinary browser appearance after “operation 2” is performed on “Anchor 1.” Figure 2 shows the behavior opening same link but using inlineLink technique. With inlineLink, several pages can be displayed simultaneously in one window.

![Figure 1: Following a link with a conventional technique](image1)

![Figure 2: Following a link with inlineLink technique](image2)

The inlineLink technique is quite simple but the reading cost is much lower than the ordinary
2.1 Open anchor and close anchors

`inlineLink` allows readers to control the insertion dynamically. Generally we call the mechanism “function anchors.” The function anchors enable readers to change linked page state, its style and appearance. The most fundamental ones are “open anchor” and “close anchor.” Open anchor loads the linked document and shows it with `inlineLink` method. Close anchor hides the inserted page.

In Figure 3, open anchors are separated from original anchors. The merit of this style is that the reader can select the link following method. If the reader clicks the original anchor, the window shows the linked document normally. If the reader chooses the open anchor, the browser turns the open anchor into close anchor, then inserts both the linked document and another close anchor as shown in Figure 4. There two close anchors functions are same.

In Figure 4, embedded document and close anchors are shown. Open anchors are separated from original anchors. The merit of this style is that the reader can select the link following method. If the reader clicks the original anchor, the window shows the linked document normally. If the reader chooses the open anchor, the browser turns the open anchor into close anchor, then inserts both the linked document and another close anchor as shown in Figure 4. There two close anchors functions are same.

Figure 3: Open anchor (Separated from original anchor)

In our early prototype, we designed to left normal anchors to open the linked page with conventional way. Though these alternatives work effectively, the label of an extra open anchor “(open)” may be an obstacle in reading. So we made eliminate version of the label by integrating into an original anchor. The example of the integrated open anchor is shown in Figure 5. The integrated style is rather straightforward and intuitive way. The reader can use a conventional anchor which appears after clicking an integrated open anchor. Figure 5 also shows the scene of recursive (hierarchical) insertion with `inlineLink` technique. `inlineLink` has no limitation in insertion levels.

2.2 Two insertion styles: wholly vs. partially

In addition to the function anchors, `inlineLink` provides two document insertion styles. One is a style which inserts whole document into the base page (Figure 6). The other is so called partial insertion which shows a part of the document (Figure 7). In most cases the whole insertion style is more effective than partial one because the reader can continue the reading task with normal scroll operations. The partial method can limit in range of displaying linked document. The latter one is convenient if the linked page is too long to display. The appearance of the
embedded window can be changed by scrolling. During the browsing, the reader can select one of these styles for each link manually.

Figure 5: Hierarchical insertion (integrated open anchor)

Figure 6: Whole insertion

Figure 7: Partial insertion
Figure 8: Rewriting Methods and translation of the contents
3 Implementation on Conventional Web Browsers

3.1 Design

In order to add inline expansion link functions into the common web browsers, we decided to utilize a page modification concept which is called “Dynamic HTML.” The “Dynamic HTML” concept consists of some other techniques. The core of the concept is a language which controls the document objects. The document objects are specified based on Document Object Model (DOM) standardized by World Wide Web Consortium (W3C). The document written in HTML is parsed and separated into some objects. These objects form a hierarchical structure can be manipulated with the instructions written in control languages such as ECMAScript.

It is so convenient design choice for both developers and readers because once the functions of inlineLink has prepared, the reader can enjoy the benefits of inlineLink with his/her favorite browsers.

3.2 Mechanism of insertion/deletion

inlineLink alters page contents by insertion/deletion. The insertion consists of two steps: (1) acquire linked document (2) display the document.

**[insertion] step 1: acquire linked document**

We use “inline frame” element which can be embedded in document. The inline frame (iframe element) is defined by HTML4.0 Transitional DTD[3]. Every HTML source can be acquired and stored in the inline frame by description `<iframe src=URI></iframe>`. By default, an inline frame is displayed on the window with specified dimension. The partial insertion is realized by this framework. In case of the whole insertion, we make the inline frame invisible. After the loading of HTML source is finished, inlineLink inserts the contents into the page.

**[insertion] step 2: display the document**

To perform the anchor rewriting and redefining of the open/close actions, we choose JavaScript. In this paragraph, we explain how to prepare an extra anchor which inserts whole document.

We should put some insertable points as span elements in the HTML source in advance. Every span element has identical label (ID) as an attribute.

Initially the contents of span element is an open anchor. When the open anchor is clicked, the action defined by the anchor invokes a function of JavaScript `insertPage()` with a target span ID as an argument. The `insertPage()` alters the open anchor into a close one. Also it inserts appropriate HTML sources after the close anchor.

**[deletion] alters anchor itself** Deletion of inserted documents is easier than insertion. Since the close anchor remembers the previous contents of open anchor, it just alters itself into the previous state.

3.3 Alter HTML Source

inlineLink is available with conventional Web browsers such as Internet Explorer if an HTML document fills these conditions.

1. An HTML source includes `insertPage()` JavaScript function.
2. “open anchors” are embedded in the HTML source.

Altering ordinary HTML pages into inlineLink version can be performed by simple rewriting filters. Thus we have prepared three conversion methods for applying inlineLink technique (Figure 8).

**Rewriting by Web Publisher in advance** We have implemented an inlineLink filter written in Perl. If the HTML source has been converted by this program, the reader can follow the link as inline.

**Server Proxy** For ordinary HTML contents, the reader can specify a proxy server which rewrites the source.
Above two methods are sometimes powerful but not convenient. Because either the publisher or the reader should pay special attention. By this method, JavaScript function automatically rewrites the loaded page before insertion. The publisher does not have to rewrite all contents, but just rewrites one anchor which links to the document root only.

### 4 Related Works

Some approaches were proposed to reduce unnecessary backtracking in browsing hypertexts. Those approaches can be classified into two categories, (a) showing additional information about linked pages near the anchor as a hint for the reader to decide whether the link should be followed, and (b) providing operations which suit to hypertext browsing for window management.

#### 4.1 Showing Additional Information

LinkPreview[4] pops up a balloon which shows the thumbnail of a linked document near the anchor when the pointer is over the anchor. HyperScout Linktool[5] takes a similar approach, that is, pops up a balloon which contains information about the linked page such as its title, author, language, last visited time, and server status.

Though these information are effective to judge the link’s value, the two approaches are not intended to support to “read” hierarchically organized hypertexts. In contrast, our inlineLink supports the reader to read several pages over hypertext hierarchies, by enabling the reader to clip two or more linked pages onto the current one.

Fluid Links[6, 7] proposes several kinds of representation to render additional information about a linked page. inlineLink shows a portable implementation technique to realize the Fluid Link's inlining without any modification to WWW browsers, though the functionality is limited currently.

#### 4.2 Opening Multiple Windows

The approach of Elastic Windows[8] gives the reader operations to open/close/replace pages in according to support typical browsing tasks. One example is opening windows each corresponding to pages that are linked from a page and shows them at once by automatically arranging the windows. Thus it provides browsability and eliminates considerable amount of step-by-step operations like “following a link and pushing `Back’ and following another link and so on.” As a result, Elastic Windows can successfully remove great number of tedious window operations.

However the reader is forced to remember the relationship between pages linked from a page shown in multiple windows. Moreover, the reader must select one appropriate operation for her/his purpose. In our approach, the relationship is explicit since every linked document is inserted directly onto the current page right after each anchor. And since the representation is simple, operations necessary to perform by the reader is only two, opening and closing.

### 5 Discussions with Informal User Study

Due to the restriction of the current filter program design and implementation, we have not finished formal evaluations yet. However, we have carried out an informal user study of inlineLink by authors and our colleagues.

We wrote a structured HTML document which includes 7 sections, one of the section has 6 subsections (in Japanese). All sections and subsections are written in separated pages. Each heading of section and subsection is an anchor which links to the appropriate contents.

Most popular feeling from users is “inlineLink makes me not to press the back button.” Usually the back button is frequently used while WWW browsing. According to Tauscher et al.[9], pressing back button is the secondary major navigating action which makes up 30 percents of all events. Thus we consider the elimination of the back action is quite effective in browsing webpages.

The most common trouble we received is
The user does not want to open all anchors because the length (height) of the document becomes longer and longer and takes much time in navigation. Especially the insertion of lengthy pages makes the user uneasiness because the index of the structured document vanishes from the window.

6 Conclusion

We presented new browsing technique inlineLink, which shows linked document by insertion. inlineLink enables readers to decrease the necessity of “back button” while browsing hypertexts. Also the reader can concentrate on the context of the current and linked webpages because redundant operations such as window size control are purged.

References


