inlineLink: Realization of Inline Expansion Link Methods on a Conventional Web Browser

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 browsing displays a web page inside of a window. In conventional web browsing, following a link replaces the previous document entirely, and the readers tend to lose the context. We have developed a system inlineLink, which applies an in-line, expansion-link method to web browsing. This in-line expansion inserts the linked document after the link anchor text. The inlineLink provides navigation mechanisms such as automatic animated scrolling, zooming, and index jumping in order to reduce the scrolling tasks while handling longer, inlined documents. We have adopted Dynamic HTML to implement the inline expansion functions. Casual users can try them on conventional web browsers. The results of our experiment prove the advantages of inlineLink in both click counts and mouse movement.

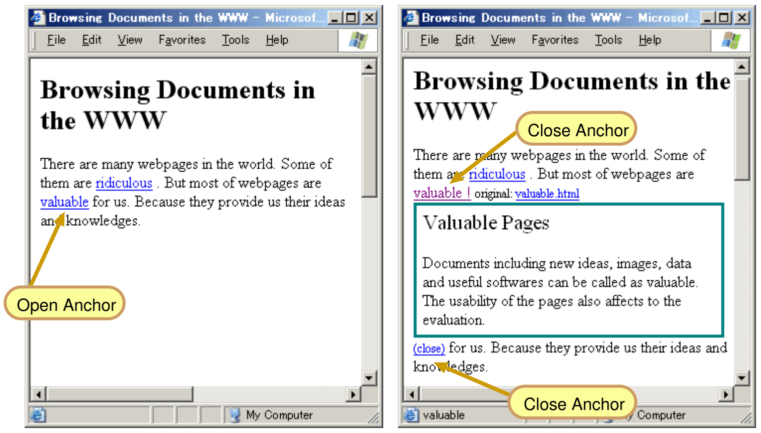
1 INTRODUCTION

Clicking on a link anchor is the most popular and fundamental operation in following links in conventional web browsers. The operation normally replaces the current document with the linked document of the window. When a reader needs to pay attention to both the current and the linked document simultaneously, the reader may choose the "open link in new window" operation to keep both documents open by window duplication. This operation is frequently selected because the linked document is closely related to the current document.

Although the open link in new window operation is effective, the duplicated window usually overlaps the current window. The reader must then change the size and location of the new, and possibly the old window, by dragging the mouse to display both documents. These operations for window management severely distract the reader reading the documents. Alternatively, the reader can drag the link anchor and drop it into another window. The operation enables the reader to specify a target window to be displayed intuitively. This operation, however, only works well if two or more windows have already been arranged on the screen. While reading the documents, the reader is still forced, however, to remember the relationships between the link anchors and the windows in which they are displayed.

2 METHOD

To solve these problems, we designed an in-line, expansion-link method and developed "inlineLink"[6], a technique that realizes the in-line, expansion-link method on conventional web browsers.


Browsing Documents in the WWW

There are many webpages in the world. Some of them are ridiculous . But most of webpages are valuable for us. Because they provide us their ideas and knowledges.

Browsing Documents in the WWW

There are many webpages in the world. Some of them are ridiculous . But most of webpages are valuable ! original: valuable.html

Valuable Pages

Documents including new ideas, images, data and useful softwares can be called as valuable. The usability of the pages also affects to the evaluation.

(close) for us. Because they provide us their ideas and knowledges.

Browsing Documents in the WWW - Microsof...
File Edit View Favorites Tools Help
My Computer
Open Anchor
Close Anchor
Close Anchor
valuable
Computer

Fig. 1. An example behavior of “activity anchors” in inlineLink

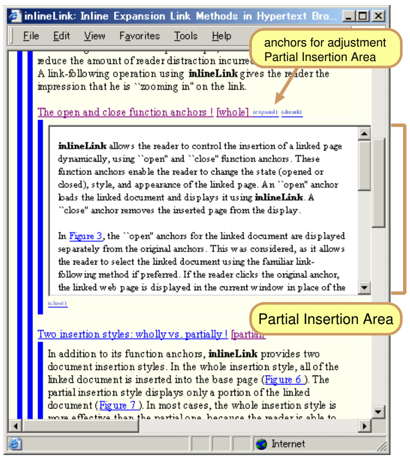
2.1 In-line, Expansion-Link Method

The in-line, expansion-link method represents a linked document. The linked document is inserted near its link anchor for easy display. In-line expansion is described as “replacement-buttons” in Guide[1], developed by Peter Brown in 1982. When a reader presses the replacement-button, the button or display is altered by the related documents. The reader can then refer to the detail of the document.

Guide handles particular hypertext contents, whereas we applied the replacement-button mechanism to the web documents written in HTML. To enable this function, we developed the “inlineLink” technique. With this technique, anchors in a normal web document are changed to “activity anchors,” which perform special functions. The primary activity anchors are “open anchor” and “close anchor.” When the open anchor (Figure 1 left) is selected, the linked document is inserted below the anchor. The open anchor then becomes the close anchor. An extra close anchor is deployed at the end of the linked document (Figure 1 right). When one of the close anchors is selected, the linked document and the extra close anchor are removed. The close anchor then reverts to the open anchor. These activity anchors enable the reader to control the appearance of the linked document.

2.2 Representation of Linked Document

In inlineLink, a representation of a linked document insertion is different from Guide, where the replaced document is embedded without any borders. Consequently, the document region is ambiguous. In inlineLink, the inserted document is surrounded by visual elements such as borders and alignments to represent explicitly the regions and their relationship to the document. The explicit representation of region and structure in the linked document makes the reader aware of its position and the relationship between the document. Because the visual elements correspond to the operation of following links performed by the reader, it may work as a history or map of the browsing activity. These histories and maps are important facilities of navigational support[7].


inlineLink: Inline Expansion Link Methods in Hypertext Bro...
File Edit View Favorites Tools Help
anchors for adjustment Partial Insertion Area
Partial Insertion Area
Internet

reduce the amount of reader distraction incurre
A link-following operation using **inlineLink** gives the reader the impression that he is ``zooming in" on the link.

The open and close function anchors ! [whole] (expand) (shrink)

inlineLink allows the reader to control the insertion of a linked page dynamically, using ``open" and ``close" function anchors. These function anchors enable the reader to change the state (opened or closed), style, and appearance of the linked page. An ``open" anchor loads the linked document and displays it using **inlineLink**. A ``close" anchor removes the inserted page from the display.

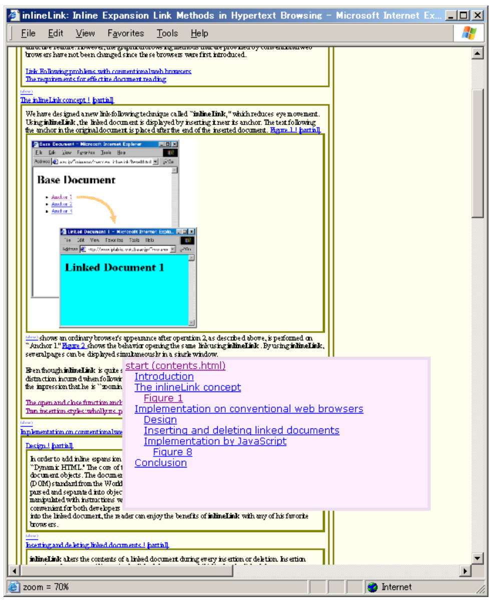
In Figure 3, the ``open" anchors for the linked document are displayed separately from the original anchors. This was considered, as it allows the reader to select the linked document using the familiar link-following method if preferred. If the reader clicks the original anchor, the linked web page is displayed in the current window in place of the

(close)

Two insertion styles: wholly vs. partially ! [partial]

In addition to its function anchors, **inlineLink** provides two document insertion styles. In the whole insertion style, all of the linked document is inserted into the base page (Figure 6). The partial insertion style displays only a portion of the linked document (Figure 7). In most cases, the whole insertion style is

Fig. 2. Partial insertion


inlineLink: Inline Expansion Link Methods in Hypertext Browsing – Microsoft Internet Ex...
File Edit View Favorites Tools Help
Base Document
Anchor 1
Anchor 2
Anchor 3
Linked Document 1
zoom = 70%
Internet

browsers have not been changed since these browsers were first introduced.

Link Following problems with conventional web browsers
The requirements for effective document reading

(close)

The inlineLink concept ! [partial]

We have designed a new link-following technique called "**inlineLink**," which reduces eye movement. Using **inlineLink**, the linked document is displayed by inserting it near its anchor. The text following the anchor in the original document is placed after the end of the inserted document. Figure 1 ! [partial]

(close) shows an ordinary browser's appearance after operation 2, as described above, is performed on "Anchor 1." Figure 2 shows the behavior opening the same link using **inlineLink**. By using **inlineLink**, several pages can be displayed simultaneously in a single window.

Design ! [partial]

into the linked document, the reader can enjoy the benefits of **inlineLink** with any of his favorite browsers.

(close)

Inserting and deleting linked documents ! [partial]

inlineLink alters the contents of a linked document during every insertion or deletion. Insertion

start (contents.html)
- Introduction
- The inlineLink concept
  - Figure 1
- Implementation on conventional web browsers
  - Design
  - Inserting and deleting linked documents
  - Implementation by JavaScript
    - Figure 8
- Conclusion

Fig. 3. Zoomed-out document view and a link structure index.

2.3 Techniques for Effective Browsing

The in-line, expansion-link method can reduce the complexity of multiple window management, but the browsing method has the following limitations.

1. The length of the embedded document influences the effectiveness of the page representation. If the document length is longer, most of the document below the open anchor is obscured.
2. The nested documents generated by repetitive insertion increase. The new, longer document may increase not only the trouble of scrolling but also navigation difficulty. Understanding the whole document structure thus becomes more difficult.

To overcome these limitations, we applied two techniques to inlineLink: (1) partial insertion and (2) navigational support.

Partial Insertion Partial insertion displays the linked document within the limited height of the embedded internal frame (see Figure 2). The partial insertion technique in inlineLink is effective in the following cases: (a) the area necessary for displaying the linked document is greater than the height of the browser window; (b) the part of the link-base document below the open anchor is crucial for the concurrent browsing task; and (c) displaying an arbitrary part of the linked page is more appropriate for the reader. The reader can change the insertion mode by selecting special activity anchors labeled “partial” or “whole.” The height of the region can also be adjusted by selecting special activity anchors labeled “expand” or “shrink” (see Figure 2).

Navigation Supports Navigation supports reduces difficulties in moving around in longer documents, and in perceiving the current position of the view port. We have prepared three functions for supporting navigation in inlineLink.

(1) Automatic adjustable scrolling function When the reader selects the open anchor in a lower view-port area, the greater part of the embedded document will not be shown on the screen. To resolve this, we introduced the automatic, adjustable-scrolling function. This function scrolls the whole document upward to make the embedded document visible after insertion. Consequently, the embedded document is placed in the middle of the view port. This function is feasible because the action of anchor selection indicates that the reader takes interest in the linked document, and will focus on the document. If the height of the embedded document is greater than the window height, the top of the embedded document is adjusted to the "ceiling" of the view port. In such cases, the inlineLink method has an advantage over conventional browsing because the reader can easily look at the relationship of the documents with a few scrolls. Without the automatic-scrolling function, the reader needs to scroll down (raise the document) frequently after selecting open anchors.

We assigned two additional operations to the embedded document. Clicking on the embedded document causes adjustable scrolling the same as the selection of an open anchor. Double-clicking on the embedded document closes the document. The former operation should lessen the reader's concerns about scrolling. The latter operation allows the reader to dismiss the embedded document instantly if the mouse is located over the document.

To reduce the cognitive efforts, the adjustable scrolling is animated. In addition to inserting the open anchors, we applied animation to the close anchors. When the close anchor is selected, the embedded document shrinks until it disappears. If the original open anchor is located above the view port, the inlineLink scrolls the document downward to reveal the open anchor. These techniques make it easier for the reader to concentrate on the document.

(2) Document-zooming function When the reader repeatedly selects open anchors, the document is nested, and becomes longer. Understanding of the longer document becomes more difficult. To alleviate the length problem, we applied the document-zooming function. The document-zooming function enables users to manipulate the zoom level of the document. Figure 3 shrinks the original document by 70%. The zoom-level manipulation is continuously performed by horizontal drag operations (right drag to expand, left drag to shrink). The zoom-level change does not affect the page layout because it preserves the position of the new line. This function helps the reader to understand the position and the structure of the document even if the window height is less than the document height.

(3) Link structure indexing function The document-zooming function helps widen the view port of the document. However, the shrink-level of the document has a boundary to read the text. In order to move the focusing point, the reader must rely on the overview of the document for finding the target position.

We designed the link structure indexing function to provide an overview for the reader. This function displays the link-structure index shown in Figure 3. The link-structure index is shown as a pop-up layer near the mouse cursor. Each item in the index indicates a close anchor, and the items are aligned to represent the structure. In most cases the label explains the linked document. Even when the label does not represent the document itself, the item identifies the anchor to the reader during browsing.

In addition to showing the structure, each item works as a link to the embedded document. When a reader selects an item, the view port of the document scrolls with an animated effect similar to the selection of an open anchor. In the current design of inlineLink, selection of the pop-up layer of the link structure index is assigned to a keyboard shortcut.

3 IMPLEMENTATION

We developed the inlineLink technique to enable both readers and page-content producers to easily browse using the in-line expansion-link method. We employed Dynamic HTML technology for enabling insertion and deletion on a conventional web browser. The document objects are specified based on the DOM[2] and standardized by the W3C. Using Dynamic HTML technology, a script written in a language such as JavaScript can handle the document based on the DOM. However, this approach may limit design for visual representation when compared with an implementation that is tightly coupled with a browser system. We regard portability and popularity as equally important when using the in-line expansion-link method.

3.1 Processes of Insertion and Deletion

The fundamental activities of inlineLink are insertion and deletion, corresponding to an open anchor and a close anchor respectively. These characteristics are implemented by the following two processes.

1. Retrieve source of the linked document
2. Replace source of the document.

It should be noted that deletion does not require retrieval, while inlineLink only requires replacement. To retrieve the arbitrary linked-document source, we utilize an "inline frame" element (`iframe`) defined by HTML 4.0 Transitional DTD[10]. To replace the source of the document, we used `insert_page()` and `remove_page()` functions in JScript. The technical details of these function are described in [6].

3.2 How to Convert inlineLink Documents

To browse an HTML document with the in-line expansion-link method, anchors in the document should be replaced with the open anchor. In addition to replacing the anchors, the document should import the inlineLink script to enable `insert_page()` and `remove_page()` functions.

One plain solution to fulfill these condition is to use a rewriting filter that converts an ordinary HTML page into an inlineLink page. However, we believe the original HTML source should be kept as it is, since the pre-conversion of the source may reduce its simplicity and maintainability. Accordingly, we have prepared the following two methods to dynamically convert an HTML source.

Script of inlineLink (by content producers) The inlineLink script is equipped with a dynamic anchor rewriting function. The built-in function converts an ordinary anchor into an open anchor before the source is pasted into the `div` element.

The content producer can provide an inlineLink-enabled page by preparing a "meta-index page." The meta-index page includes an open anchor to the original index page.

The content producer does not have to convert the original anchors in either the index page or the other contents. The readers can choose their favorite type of document by entering either the original index page or the meta-index page. If the reader chooses the meta-index page, the anchor tags in the linked page are automatically replaced as open anchors. The replacement is a trivial task performed as a local process. The replacement is performed recursively so that the reader can continue browsing with inlineLink.

On-demand conversion with inlineLink servlet (by readers) The method in the above section (3.2) applies when the content producer has prepared the meta-index pages. If no meta-index page is presented, the reader cannot utilize the in-line version of the content.

We have implemented an inlineLink Servlet that converts anchors in the HTML source from an arbitrary site into open anchors. The converter Servlet (inlineServlet) obtains the source of the requested URI. Then it parses the source with our HTML parser. The inlineServlet converts ordinary anchors into open anchors, and replaces URIs in each anchor as requests for the inlineServlet. As a result, all requests from anchors converted by the inlineServlet go through the inlineServlet. Since an inlineServlet is designed to work with a common servlet container, the reader can install an inlineServlet at any computers including a PC.

With the inlineServlet bootstrap page, the reader only needs to specify the URI in the form and press the [convert] button. The reader can continue browsing with in-line expansion-link method because the page content of each request is converted by the inlineServlet. Furthermore, the reader can bookmark the URI including the request to the inlineServlet.

4 EXPERIMENTS

To measure the effectiveness of the inlineLink in page browsing tasks, we performed a usability and performance study on 12 subjects. We conducted the usability test on an 850 MHz portable notebook PC powered by Windows 2000, with 384MB RAM, XGA display (1024$\times$768 pixels), and a two-button mouse with a wheel.

Microsoft Internet Explorer (IE) 6.0 performed all tasks. We maximized the IE window, showing the standard buttons. We then compared the following three conditions.

- [normal] is the basic feature of the IE. Subjects were allowed to use “back” and “forward” button.
- [inline] is the in-line expansion-link method. Subjects used the inlineLink without an automatically adjustable scrolling function.
- [inline (adjust)] is similar to the [inline], but subjects browsed the inlineLink document with the automatic adjustment scrolling function.

To record the data, we developed a tiny counter tool[1] that hooks Windows’ system events. We measured (1) the number of mouse button clicks, (2) the distance of mouse-pointer move, (3) the distance of the mouse pointer drag, (4) the amount of wheel rotation, and (5) working time.

[1] http://www.iplab.is.tsukuba.ac.jp/˜miuramo/wheelcounter/

Experiment 1 (Glossary) We prepared a CGI-based glossary site to evaluate the inlineLink while browsing relatively short web documents. The glossary includes computer and internet terms. An anchor represents the inlineLink term in the glossary, which is written in Japanese. We asked each subject to answer 12 questions. Each question is designed so that the subject must refer to at least two explanations of terms. An example question is "Which organization was established earlier, ISO or IEC?" The subjects start with a page that includes the questions and links to the related terms. The height of the document, which inserts all related terms, is up to six times greater than the height of the browser window.

Experiment 2 (StyleGuide) To measure the effects of a greater number of documents, we employed a Japanese translation of "Style Guide for Online Hypertext" authored by Tim Berners-Lee[2]. The document consists of 28 pages, 23 of which can be followed by index. Others are supplemental topics followed by a description of the document. The height of the document, which inserts all document, is about 50 times greater than the height of the browser window.

We prepared 15 questions divided into three subtasks. The first 10 questions simply require finding the sentences in lines. The last five questions require a detailed understanding of the document. An example question is "What is the ideal length of a title in characters?" A subject must answer by referencing the sentence in the document, "The title should be less than 64 characters in length."

Observations and results First we described the activities of subjects after observation. During the test, the height of the working window with inlineLink was two to three times greater than the window height in the Glossary. In the StyleGuide, the height was 10 to 20 times greater than the window height.

In the [normal] method, subjects were allowed to utilize any mouse operations including open link in new window, move/resize of window, and drag the anchor and drop it to another window. As a result, half of the subjects chose the "open link in new window" operation in the Glossary. The subjects who opened one window and dropped anchors could effectively finish the tasks. The subjects who opened more than one window took much more time in management tasks such as moving, resizing, and overlapping. Some subjects lost the original window while handling other similar windows.

Table 1 indicates the result of the paired t-test ($p = 0.05$). The t-values without parentheses indicate the significance between two compared methods. The "$+$" mark represents results that show the right-hand method is better (less) than the left-hand method, whereas the "$-$" shows the opposite.

In the Glossary task, both [inline] and [inline(adjust)] significantly reduced the mouse click count ($t = 4.52, p = 0.05$) and the distance of the mouse move ($t = 3.47, p = 0.05$). The reason may depend on the ratio of linked-document height to window height. In the Glossary task, few subjects closed the inserted documents.

In the StyleGuide task, only the [inline] method reduces the distance of the mouse move significantly ($t = 2.12, p = 0.05$). One of the reasons why [inline] did not decrease the mouse click count in the StyleGuide is that some participants frequently used

[2] http://www.kanzaki.com/docs/Style/ (translation)
http://www.w3.org/Provider/Style/ (original)

Table 1. Result of paired t-test

Experiments	methods compared	mouse click	mouse move	time	wheel up	wheel down	mouse drag	significant border (5%)
Glossary	normal - inline	+4.52	+3.47	(+1.65)	-2.40	-4.37	+3.05	1.80
Glossary	normal - adjust	+4.62	+3.31	(+1.15)	(-1.60)	(-1.72)	+2.96	1.80
Glossary	inline - adjust	(+0.32)	(+0.71)	(-0.46)	(+0.53)	+2.12	(-0.89)	1.80
StyleGuide	normal - inline	(+0.93)	+2.12	(+1.12)	(-0.36)	(+0.62)	(+1.74)	1.80
StyleGuide	normal - adjust	(-0.26)	(+1.26)	(+0.04)	-2.52	(+0.30)	+2.38	1.80
StyleGuide	inline - adjust	(-1.38)	(-1.66)	(-1.41)	-2.06	(-0.45)	(+1.30)	1.80

"closing by double click" instead of using the close anchor. In the [inline(adjust)] task, the automatic scrolling function may generate a gap between the mouse pointer and the close anchor. The gap encourages double clicking and discourages selection of the close anchor.

Although the average working time tended to decrease with the inlineLink, the data does not illustrate the significance of using our technique. In our experiment, each task completion time included some extra working time such as reading, understanding and forgetting, besides the pure operating time. The extra working time may distort the effects.

In the StyleGuide task, the [inline(adjust)] method significantly increased the mouse wheel rotation ($t = 2.52, p = 0.05$). The reason of this phenomenon is that the subjects tried to scroll upward against the automatic scrolling. In the Glossary task, the wheel rotation significantly increased (upward $t = 2.40$, downward $t = 4.37$) with the [inline] method. This result does not indicate the disadvantage in the inlineLink method because the Glossary document is too short to display the scroll bars with the [normal] method. In the StyleGuide document, the wheel rotations are frequently used even in the [normal] method. The amount of the downward wheel rotation with the inlineLink method is restrained to minimize the significance. As an exception, the upward wheel rotation increased with the [inline(adjust)] method in the StyleGuide task. The result derives from the insufficiency of the automatic scrolling function. The downward scrolling at insertion is straightforward, whereas the upward scrolling at deletion has not been tuned for these experiments, so the subjects should perform some wheel-up operations after the deletion.

To summarize the results of the experiments, the in-line link method significantly reduced the mouse click counts and the distance of the mouse movements where the embedded documents were short. The automatic scrolling function also reduced the amount of downward wheel rotations.

After the experiments, we interviewed the subjects. Some subjects commented that the automatic upward scrolling at deletion is necessary. Most of the subjects answered that "double click to close" is feasible, but a few subjects said that there is an alternative for mapping close functions. The comments regarding the effectiveness of the inlineLink include "useful if the embedded document is shorter" and "I might switch from [normal] to [inline] depending on the situation."

5 RELATED WORKS

LinkPreview[5] produces a pop-up balloon window to display a thumbnail of the linked document near the anchor when the pointer is over the anchor. HyperScout Linktool[9] takes a similar approach. It produces a pop-up balloon window that contains information about the linked page, such as its title, author, language, time last visited, and server status. Neither of these approaches supports reading of hierarchically organized hypertexts. In contrast, the inlineLink technique allows readers to read multiple pages in a hypertext hierarchy by inserting two or more linked pages into the current one.

Fluid Links[11, 12] proposed several kinds of displays to add information about a linked page. The inlineLink uses a portable implementation technique that produces an effect similar to Fluid Link's *inlining* without any modification to the web browser.

SmallBrowse[8] is a Programming by Example (PBE) system that suggests a link to be selected by considering the reader's web browsing history. The targeted display of this system is a small display such as a PDA. SmallBrowse reduces the selection tasks by indicating predicted links as pop-up windows called "tip help." The objective of the SmallBrowse is similar to that of inlineLink. SmallBrowse works effectively, especially for recurrent browsing, whereas inlineLink can control the granularity of the browsing information specialized for structured documents.

The Elastic Windows[3, 4] method allows readers to open, close, or replace pages as required to support typical browsing tasks. For example, a single operation can open multiple windows, each corresponding to a link on the current page. All of the windows are displayed simultaneously, and the placement and size of the windows are automatically set as part of the operation. This operation allows the reader to browse effectively, while at the same time eliminating a considerable number of step-by-step operations such as "following a link" and selecting the "Back" or "Forward" buttons. Elastic Windows thus successfully removes a significant number of tedious window operations. However, readers have to remember the relationship between the pages that are shown in the multiple windows, and the reader can only select the one operation that is most appropriate for him. In our approach, relationships are explicit, since every linked document is inserted directly into the current page right after its anchor. Also, since the representation is so simple, the reader only has to choose between the opening and closing operations.

6 CONCLUSIONS

We described the in-line, expansion-link method and inlineLink, which applies the method to commonly used web documents. Using the inlineLink significantly reduces the use of the "Back" button while browsing a web document is. The feature is quite simple but powerful and effective in enabling readers to concentrate on the context of the web document itself. As the inlineLink is designed to work on conventional web browsers and web documents, the readers can easily apply the interface for their browsing tasks. The empirical study revealed the advantage of the inlineLink that can reduce both the number of mouse button clicks and the distance of the mouse movement.

The inlineLink scripts and servlets are available from the following URI.
http://www.iplab.is.tsukuba.ac.jp/˜miuramo/inlinelink/

References

1. P. J. Brown. Turning Ideas into Products: The Guide System. In *Hypertext'87 Proceedings*, pages 33–40, Nov. 1987.

2. Document Object Model (DOM) Level 2 Specification. (Web Page), May 2000. `http://www.w3.org/TR/DOM-Level-2/cover.html`.
3. E. Kandogan and B. Shneiderman. Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser. In *Proceedings of the 10th annual ACM Symposium on User Interface Software and Technology (UIST'97)*, pages 169–177, Oct. 1997.
4. E. Kandogan and B. Shneiderman. Elastic Windows: Evaluation of Multi-Window Operations. In *conference proceedings on Human factors in computing systems (CHI'97)*, pages 250–257, Mar. 1997.
5. T. Kopetzky and M. Mühlhäuser. Visual Preview for Link Traversal on the WWW. In *Proceedings of the 8th International World Wide Web Conference (WWW8) / Computer Networks 31 (11-16)*, pages 1525–1532, 1999.
6. M. Miura, B. Shizuki, and J. Tanaka. inlineLink: Inline Expansion Link Methods in Hypertext Browsing. In *Proceedings of 2nd International Conference on Internet Computing (IC'2001)*, volume II, pages 653–659, June 2001.
7. J. Nielsen. *Multimedia and Hypertext: The Internet and Beyond*, chapter 9, pages 247–278. AP Professional, 1995.
8. A. Sugiura. Web Browsing by Example. In H. Lieberman, editor, *Your Wish is My Command – Programming by Example –*, chapter 4, pages 61–85. Morgan Kaufmann Publishers, 2001.
9. H. W. R. Weinreich and W. Lamersdorf. Concepts for Improved Visualization of Web Link Attributes. In *Proceedings of the 9th International World Wide Web Conference (WWW9) / Computer Networks 33 (1-6)*, pages 403–416, 2000.
10. World Wide Web Consortium (W3C). HTML 4.01 Specification. (Web Page), Dec. 1999. `http://www.w3.org/TR/html401/`.
11. P. T. Zellweger, B.-W. Chang, and J. Mackinlay. Fluid Links for Informed and Incremental Link Transitions. In *Proceedings of HyperText'98*, pages 50–57, 1998.
12. P. T. Zellweger, S. H. Regli, J. D. Mackinlay, and B.-W. Chang. The Impact of Fluid Documents on Reading and Browsing: An Observational Study. In *Proceedings of the CHI 2000 conference on Human factors in computing systems (CHI'00)*, pages 249–256, Apr. 2000.