

A Study on Web Document's Efficient Browsing

Dong-Hyun Kim, Seung-Heon Song and Eung-Kon Kim, *Member, KIMICS*

Abstract—Most document consists of primary content and supporting material, such as footnotes, detailed explanations, and illustrations, and the related supporting materials are linked as hypertext on web document. However, the content of hypertext links is appeared in the new windows on present web browser. Then the user will leave the primary material, may lose the entire context, and must have some difficulties to return to the primary context when the interest disappears. Using the technique for fluid links, we can solve these problems easily. If the mouse is putted on the link, the related material is presented in between lines or at margin maintaining the context of primary material. In this paper, we introduce the various browsing techniques using fluid links, analyze the forms and the features, and then we propose the way to implement in Java.

Index Terms—Web document, Fluid link, Hypertext, Java Language, Computer Graphics

I. INTRODUCTION

Emergence of WWW brought Internet into widespread use and the number of Internet users is explosively increasing. Web browser has been used as a common tool for the efficient use of Internet and has been delivering enormous information from Internet to its users in convenient ways [1].

Hypertext is the most distinguished feature of the documents used in Web browser. Hypertext enables the users to directly reach for the needed information through its link. However, it may cause the “disorientation” – unintended change of the main screen to its “cognitive overhead” – unintended losing track of main document [2].

It apparently reveals a need for more efficient communication method between web-page providers and information users, and for new way of approach for its solution.

example, in currently used web pages, users have to move to a certain hypertext or certain space for explanation in order to obtain detail information for certain hypertext or word. This means that users have to keep moving

from one HTML document to another by pushing [Forward] and [Previous] keys.

In this thesis, we introduce several browsing methods using Fluid Link and also their structure, and discuss the results of its actual visualization by Java language.

We have implemented some versions of the fluid link technique, including interline expansion, margin callout, textual overlay, and annotations displayed in multiple layer.

II. RELATED WORK

In this chapter, we introduce several methods of browsing information and documents from several systems on the WWW.

Since its advent in 1954, hypertext has been playing its important role as a link connecting more than two information in different locations, and enables users to move from one physical point to another.[3]

However, users must move to a new anchor in order to obtain information from that hypertext. Although Internet Explorer Version 4.0 made it possible to view brief information on the anchor such as its name or keyword by placing mouse on it, users still have to move to new anchor in order to gain detailed information and view its main text.[4] Fluid Link method provides the solution for this problem.

Overlapping method put optical relevance among individual factors on the space of information into diagram. It was used only for the visual indication of several related information from The NiF Elastic Catalog. Fluid Link is used for dividing linked documents from main text in content as well as for visual indication.[5]

Zooming method magnifies and brings the chosen information to the front while fading the original screen away simultaneously from Pad++ Browser.[6]

Focus+Context method enables users to seize the point of information as well as provide a good grasp of the details at the same time.[7]

With Fluid Link method, users can get the same or even better result with Focus+Context method by maintaining direct control over the size and appearance of the main text. Fluid Link method helps users to avoid unnecessary disorientation or cognitive overhead.

III. BROWSING METHODS APPLING FLUID LINK

3.1 Displaying annotation of blank space

Annotation will be placed on left, right or right below the anchor wherever the user plotted for annotation.

Manuscript received April 17, 2003.

Dong-Hyun Kim is with the Department of Computer Information, Suncheon Cheong-am College, Suncheon-si, Korea.

Seung-Heon Song is with the Graduate School of Computer Science, Suncheon University, Suncheon-si, Korea.

Eung-Kon Kim, corresponding author is with the Department of Computer Science, Suncheon University, Suncheon-si, Korea. (Phone: 061-750-3627, e-mail: kek@suncheon.ac.kr)

Users can view the annotation on left or right side of main text by placing or clicking mouse on it as shown in Figure 1.

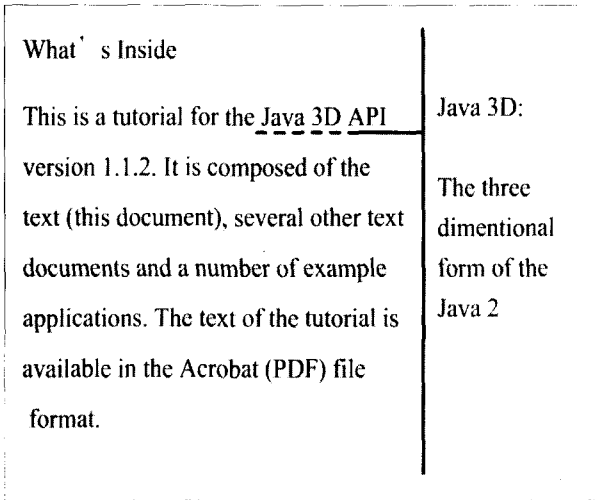


Fig. 1 Displaying annotation of blank space

It is also possible that users can place the annotation between lines if required for better appearance of the text as shown in Figure 2.

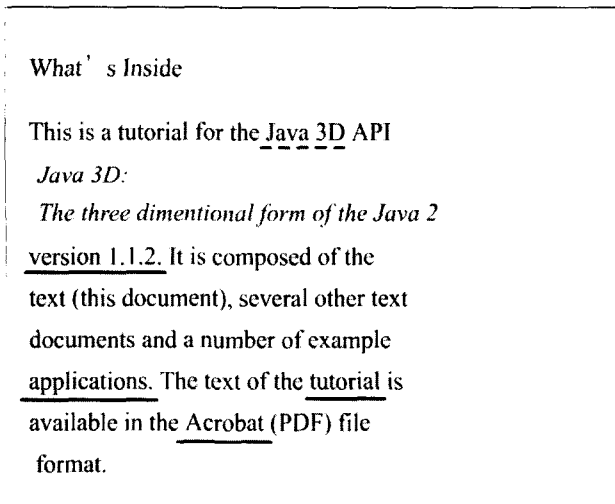


Fig. 2 Display annotation of blank space

3.2 Placing the annotation over the main text.

This method has a thread of connection with layer method in other graphic programs. Users can view the text of linked documents or annotations in semi-transparent cell type by placing the mouse pointer on the targeted anchor. Semi-transparent cell typed – different color from main text – annotation will be visible regardless of the changes given to the main text. The source material in that space does not move away; rather, it fades in color. The difference in darkness of the two layers of text, as well as other differences like size and font, can make the supporting material readable while still preserving some degree of readability of the underlying source text like in Figure 3 [3].

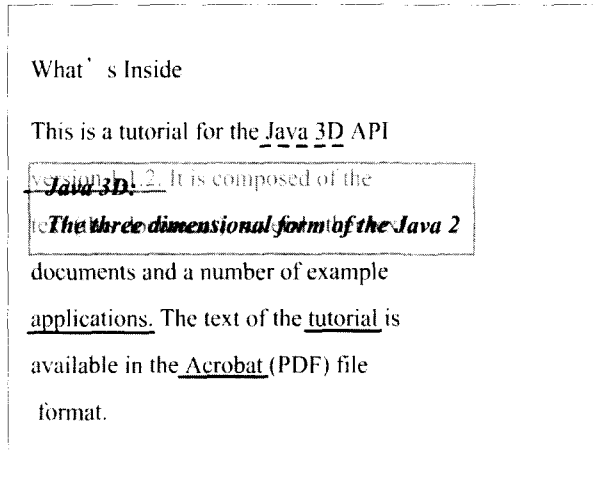


Fig. 3 Placing the annotation on top of the main text

3.3 Annotating method applying high resolution system

This method shall be the most efficient way of annotating provided that the system supports two or three times of high resolution. Because it can be annotated 2 or 3 points letter in between main text without giving any change to its original feature like in Figure 4.

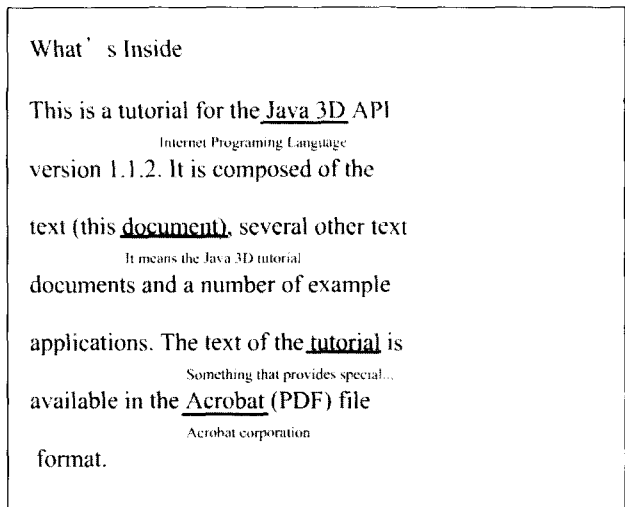


Fig. 4 Placing the annotation on top of the main text

3.4 Annotating method displayed in multiple layer

Further to referred methods, combining Figure 1 and Figure 3 shall be proposed for closer and clearer appearance of annotations displayed in multiple layer. As shown in Figure 5, this method can annotate while maintaining the original structure and put additional annotations for existing annotations on blank space.

It requires consistent standard for the location of overlaid annotation that would be placed on the main text and additional annotations that would be placed in blank space.

The method referred in Figure 5 will provide users more detailed information by enabling them to put additional annotation to existing annotations. These annotations include the editor, modified date, URL, meta data and other related information of the document [8].

What ' s Inside

This is a tutorial for the Java 3D API version 1.1.2. It is composed of the **Java 3D:** ~~(The three dimensional form~~ other text of the Java 2 documents and a number of example applications. The text of the tutorial is available in the Acrobat (PDF) file format.

Dimention:
A measure of spatial extent and so on...

Fig. 5 Simultaneous use of Figure 1 and Figure 3

IV. IMPLEMENTATION OF FLUID LINK

We used Java as a programming language to implement Fluid Link in this paper [9]. Programs made only by library can be performed in any kind of computers because Java is structurally neutral by adopting byte code.

Users can have direct control over the files and objects at remote location and are provided with methods to collaborate with programs at remote locations because Java applet program is related with WWW, receives information from server, and carries them to client.

Higher reliability will be required in environment where more than two computers are connected to one program for efficient performance.

Java excluded many possible problems that may occur on the basis of pointer – often found in other languages such as C, C++ – by not providing certain pointer. Java also performs automatic garbage collection for removing useless memory.

Using Java applet, the expression of main text on the web browser will be visualized to the transparency of annotations. New type tag <Fluid> should be inserted to express annotations, new type documents called FHTML is presented for the description of interaction between HTML documents and applet.

Applet can be executed by calling in HTML on the Web browser without any command. HTML documents loaded on the browser call the applet in the server, and the applet can be executed.

Client/Server environment is established by the use of socket provided by Java and java.net.Socket, java.net.ServerSocket calss in order to use Java applet.

```
S= new Socket(getCodeBase().getHost(),2504)
/**
the example of the method to generate socket in client
and to connect specified host name and port(2504)
**/
```

Fig. 6 An example for use of the Socket method

Socket class makes socket needed by the client to request to server and provide consecutive stream. Server socket class is the socket used for the communication. [Figure 7]

```
ServerSocket ss = (ServerSocket)null;
|
ss=new ServerSocket(2504)
|
soc = ss.accept();
createSocketThread();
soc.close();

/** an example of the server socket to supervise if
there is a requirement from client **/
```

Fig. 7 An example for use of the ServerSocket method

This Client/Server environment is used for the interaction among computers and also builds security.

Figure 8 shows the HTML document type for the implement of applet in HTML document. <APPLET> </APPLET> tag shall be inserted in HTML document for the use of applet.

```
<html>
<head>
<meta name = "generator" content = "ModelWorks
IDE">
<title>Fluid</title>
</head>
<body>
<h2>Test page for Fluid Class</h2><hr>
<applet code = Fluid.class width = 300 height = 300>
</applet>
<hr></body>
</html>
```

Fig. 8 The imple mentation of applet in HTML document

```
<FHTML>
<head>
<title>Fluid Link Document Example</title>
</head>
<body>
<H2> What's Inside </H2> <BR>
This is a tutorial for the Java 3D API< BR >
version 1.1.2. It is composed of the< BR >
text (this document), several other text< BR >
documents and a number of example< BR >
applications. The text of the tutorial is< BR >
available in the Acrobat (PDF) file< BR >
format.< BR >
</body>
</FHTML>
```

Fig. 9 An example for use of the FHTML document

Figure 9 shows the HTML document type applying Fluid Tag, FHTML is designed for the application of new type Fluid tag to HTML documents. <Fluid></Fluid> tag shall be inserted for the use of Fluid tag. [Figure 10]

```
<Fluid Type=overlay Src=introduce.fhtml>
  Java3D
</Fluid>
/**Adjust of the Fluid Tag:
  overlay - to lay or spread over the text
  Src - reference of the annotation (fhtml
document )
**/
```

Fig. 10 An example for use of the Fluid Tag

MouseEvent is a method used for the initiation of Fluid link with mouse event in Java applet, it will be called when mouse cursor enter into the range of applet. Coordinate number should be required. [Fig 11]

```
public boolean mouseEnter
(Event e, int x, int y)
  display Fluid(this.type, this.fhtml)
  return true;
/** The method of run by mouse over event **/
```

Fig. 11 An example for use of the mouseEnter method

Java provides java.awt(Abstract Window Toolkit) package for the construction of GUI (Graphic User Interface). Figure 12 shows that PixelGrabber class from the package will make it possible to designate the location and the size of the annotation display on Java applet.

```
Import java.awt.*;
Import java.awt.image.PixelGrabber;
Public class Fluid extends java.applet.Applet
{
private int [] resolveImage (Image Java 3D, int
width, int height)
{
int pixel [] = new int[3--*100];
Pixelgrabber pg =
New PixelGrabber(Java 3D, 0, 0, width, height,
pixel, 0, 300);

Try {
Pg.grabPixels();
}
catch(Exception e)
{System.out.println(e.toString());}
return pixel;}
}
```

Fig. 12 An example for use of the PixelGrabber class

In this paper, we develop the browser for Fluid link using JDK 1.2.2 in Windows 95. Figure 13 shows the result of implementing the browser for Fluid link documents.

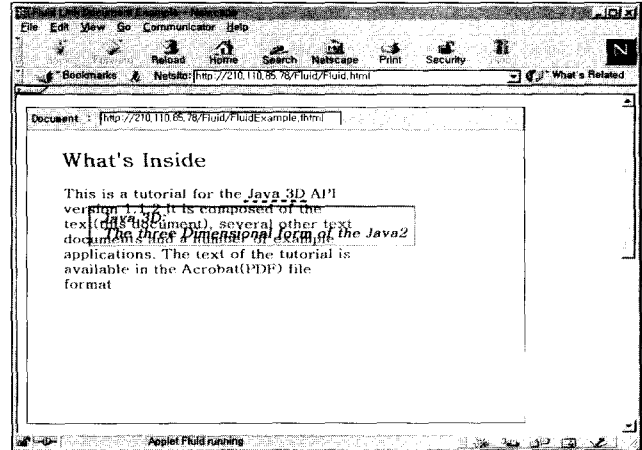


Fig. 13 The result of implementing the browser for Fluid link document

V. CONCLUSION AND FUTURE WORK

We have introduced the several browsing methods applying Fluid Link for more efficient navigation on the WWW, have compared with their features and advantages, and have shown how to implement the browser using Java language.

So far, WWW users have not been receiving sufficient support from information providers. Though hypertext is useful, the user will leave the primary material, may lose the entire context, and must have some difficulties to return to the primary context when the interest disappears. Using the technique for fluid links, we can solve these problems easily. If the mouse is putted on the link, the related material is presented in between lines or at margin maintaining the context of primary material.

In this paper, we provided WWW users with easier and quicker approach to required information by enabling them to omit unnecessary steps in searching information as a result of applying Fluid Link method in Java language. We have implemented some versions of the fluid link technique, including interline expansion, margin callout, textual overlay, and annotations displayed in multiple layer.

This browser for Fluid link will help information providers to deliver consistent and clear information in more efficient ways. In the future work, we will extend to applying to multimedia document such as images, sound, movie. And the interaction of users and the linked information need to be improved.

REFERENCE

- [1] Nielsen, J, "The tyranny of the page: continued lack of decent navigation support in version 4 browsers", <http://www.useit.com/alertbox> (1997).

- [2] Bieber, M., Vitali, F., Ashman, H., Balasbramanian, V., and Oinas-Kukkonen, H., "Fourth generation hypermedia: some missing links for the World Wide Web", *Int. J. Human-computer Studies*, Vol. 47, pp. 31-65 (1997).
- [3] Polle T. Zellweger, Bay-Wei Chang, Jock D. Mackinlay, "Fluid Links for Information and Incremental Link Transitions", *Proceedings of Hypertext'98*, Pittsburgh, PA, 20-24, pp. 50-57 (June 1998).
- [4] Bush, V., "As We May Think." *The Atlantic Monthly*, 176, pp. 101-108 (July 1945) Reprinted and discussed in *interactions*, pp. 35-67, 3(2) (Mar 1996).
- [5] Murtaugh, Michel., "NiF Elastic Catalog", <http://tablesaw.media.mit.edu/ecat/> (1996).
- [6] Bederson, Benjamin B., James D. Hollan, Jason Stewart, David Rogers, Allison Druin, and David Vick. "A zooming web browser", *Proc. SPIE Multimedia Computing and Networking'96*, Volume 2667, pp.260-271 (1996).
- [7] Young-a Jung, "Implementation of a Zoom Browser using the Focus+Context technique", pp. 9-12 (Feb 1999).
- [8] Dominic Stanyer and Rob Procter, "Improving Web usability with the link lens", *Computer Networks*, 31, pp. 1533-1544 (1999).
- [9] Sun microsystems, "Java Language overview - white page", <http://java.sun.com/docs/overviews/java/java-overview-1.html> (1999).



Dong-hyun Kim

Received B.S. degree in computer engineering from Chosun University, Kwang-ju, Korea, in 1986, the M.S. degree in Computer engineering from Gwang-woon Univ. in 1992, and the Ph.D. degree in Computer Science from Chosun Univ. in 2002.

He worked for Honam Branch, HyunDae Electronics Co. as a chief executive from 1989~1996.

Since 1996, he is currently an associate Prof. of Dept. of Computer Information, Suncheon Cheong-am College, Suncheon, Korea.

He is interested in Multimedia, EC, Digital Contents & its applications.

Dr. Kim is a member of the KIMICS.



Seung-heon Song

Received B.S degree in Department of Physics from Seonam University, Namwon, Korea in 1998 and M.S. degree in Major of Industrial Information Management The Graduate School of Information Science, Sunchon National University, Sunchon, Korea,

in 2000.

He is currently a Ph.D. candidate in Department of Computer Science, Sunchon National University.

He is interested in Computer Graphics and Multimedia.



Eung-kon Kim

Received the B.S. degree in electronic engineering from Chosun University, Gwangju, Korea, in 1980, the M.S. degree in electronic engineering from Hanyang University, Seoul, Korea in 1987 and the Ph.D. degree in Computer engineering from Chosun University

in 1992.

He is currently a professor of Department of Computer Science, Sunchon National University.

He is interested in Computer graphics and its applications.