An Online Method for Editing Any Web Page Using Proxy Agents

Tadachika Ozono,[†] *and Toramatsu Shintani*[††],

Computer Science and Engineering, Graduate School of Engineering
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555 JAPAN

Summary
We propose an online method WFE for editing any Web page by realizing a proxy agent. The main strength of WFE is that the method can provide users with a way to edit any web page regardless of its original author. When a user access a web page via the proxy agent, the proxy agent augments the web page with client-side scripting via JavaScript that provides the user with the editing functions. By utilizing the XHTML DOM, the script included by the proxy agent allows the user to simply select any portion of text on the page and then directly edit the underlying HTML even if the selected text appears in many different places in the document. WFE identifies the specific instance of the selected text to be found. We also describe the system called WPM to show how effectively the method WFE can be used. WPM provides marking and anchoring functions on ordinary web browsers. The users can mark words and phrases on web pages by using their browsers without any extra plug-ins like similar systems. The result shows that the method WFE is effective in developing Web applications such as the system WPM.

Key words:
WWW, Dynamic HTML, Web Page Edit

1. Introduction

The World Wide Web (WWW) has become an important information transmission environment. WWW enables users to publicize information on their Web pages. However, making and editing Web pages are still bothersome issues for novice users. Although Web authoring systems (e.g., Dreamweaver, Go-Live, etc.) can help, the systems require a certain amount of knowledge for effective use. In this paper, we propose a new online method called WFE (**W**eb **F**lexible **E**diting) for making and editing any Web page. WFE can provide users with a way to edit any web page in which users can edit and delete existing texts and add new texts and images on a Web page by using a proxy agent. Users of WFE can even add handwriting graffiti on Web pages. When a user access a web page via the proxy agent, the proxy agent augments the web page with client-side scripting via JavaScript that provides the user with the editing functions. By utilizing the XHTML DOM, the script included by the proxy agent allows the user to simply select any portion of text on the page and then directly edit the underlying HTML even if the selected text appears in many different places in the document. WFE identifies the specific instance of the selected text to be found.

Users can edit any Web page in which WFE does not directly edit an HTML text for a page on a Web server. In WFE a user can simply edit an HTML text cached by a proxy agent. The proxy agent also enables us to modify dynamically generated Web pages or Web pages that get updated continuously. The proxy agent can save the modified page (an HTML text) to a Web server to publicize the modified page via the Internet. WFE does not require particular any extra plug-ins or software components like similar systems. Users can edit Web pages using common existing Web browsers (e.g., Internet Explorer, Safari, Mozilla). Although the DOM interfaces are not truly standardized across browsers, WFE works for the common Web browsers because WFE provide some functions that depend on the DOM interfaces of the Web browsers.

To show how effectively the method WFE can be used. We have implemented the system called WPM (Web Page Marker), which provides marking and anchoring functions on ordinary web browsers. WPM is designed to realize a effective system, which is one of Web browsing support systems [9]. WPM users can mark words and phrases on web pages by using their browsers. There are Google Toolbar [6] and a Mozilla Firefox [4] as an existing tool with a similar function. Google Toolbar has a function which indicates a word used for reference by highlight. Firefox is a Web browser with an emphasis display function in the reference in a page. WPM doesn't require any preparing plug-in and special software components by using WFE.
The rest of this paper is organized as follows. We first show an outline of the WFE. Then, we describe the implementation of WFE. Here, we explain about the proxy agent for WFE and the dynamic editing function based on JavaScript. Next, we show the outline of WPM. Finally, we discuss the features of the method, and then we describe some concluding remarks.

Manuscript received May 5, 2006.
Manuscript revised May 25, 2006.

2. An outline of WFE

There are several existing systems to post information using a Web browser. BBS (Bulletin Board System), which is often implemented as a CGI program, is one way to post information. In BSS, since users must follow its format to input text and images, users cannot freely edit Web pages. Wiki [7][12] is an administration system for Web content for collaborative Web page construction. Although users of Wiki can edit HTML texts and generate Web pages using a Web browser, they must know its original scripts and tags to do so effectively. On the other hand, WFE does not require such format and original syntax. Moreover, it can be used for every Web page on the Internet. It has several technical advantages as follows:

- Users simply indicate the editing action (e.g., insert text, add comments) for the target Web page. Unlike other technologies, WFE does not need particular Web page forms to input data or texts on Web pages. When a user is browsing Web pages and wants to post information, he/she simply select the text by mouse dragging and clicking the place where the text is to be placed on browser. It is almost like putting a Post-it (sticky note) on paper.

- There is no need to reload the Web page to see the results. Reloading an edited page would be bothersome, particularly, if it is a large Web page with much text and many images. With WFE, users do not need to reload after editing. When they edit a Web page, the results are reflected immediately. For example, when a user edits a sentence on a Web page, the sentence is immediately replaced with the new one.

- This pseudo-editing is done by using a proxy agent. Users browse and edit Web pages via the proxy agent. When a user edits a Web page, the edited data is stored into a cache memory managed by the proxy agent. The proxy agent can merge the original Web page and the edited data, and then generates a new Web page reflecting the user's edited data. Accordingly, although the original Web page is not modified, users can browse edited it via the proxy agent. Users not browsing via the proxy agent will simply view the original one.

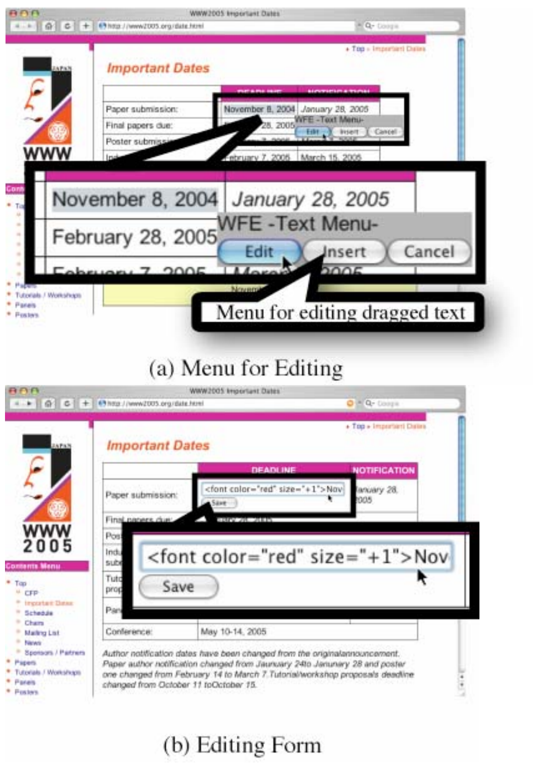


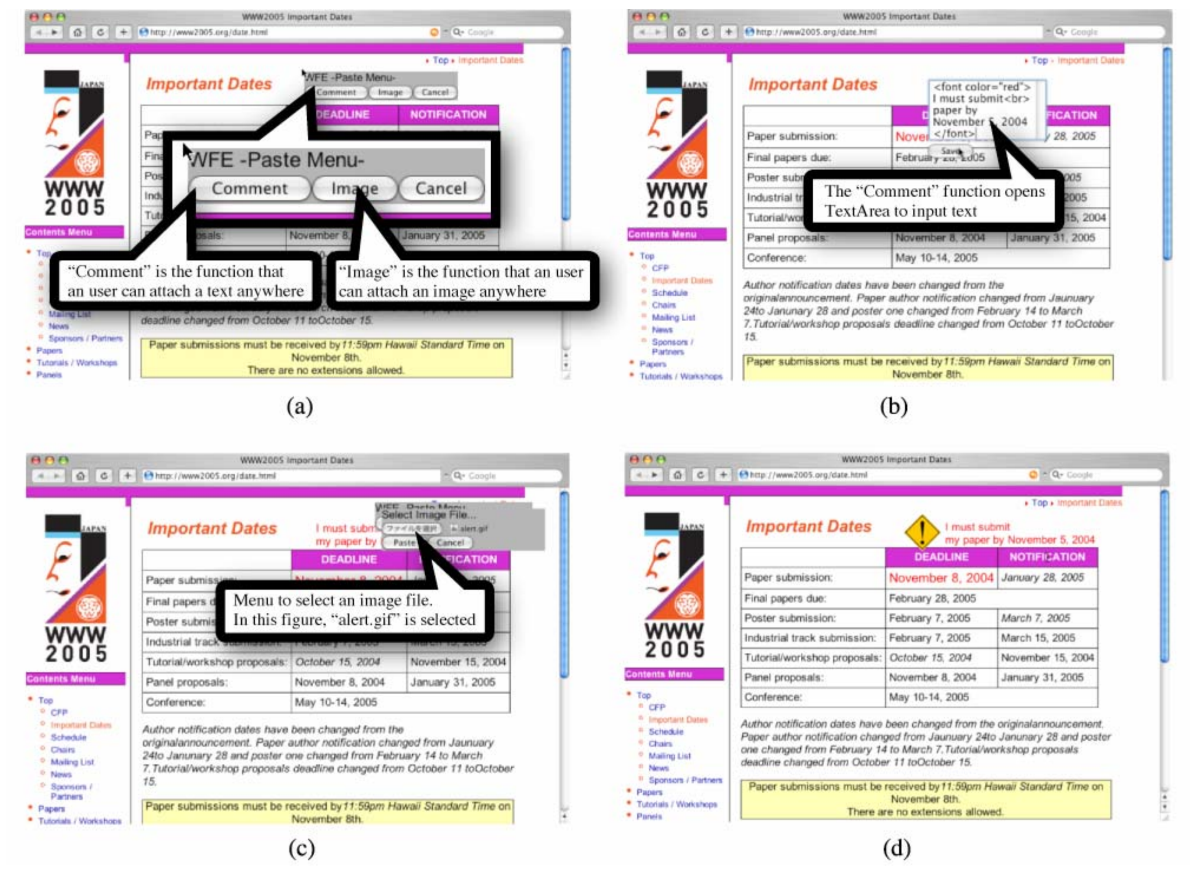
Fig. 1. Editing a Web Page using WFE

Figure 1 shows an example of editing a Web page using WFE. The user wants to modify a sentence describing a submission deadline for WWW2005 (http://www2005.org/papers/). First, the user selects the target text, "November 8, 2004", by mouse dragging in the Web browser. Next, the user clicks the right mouse button to open the WFE menu (Figure 1-(a)). When the user clicks the "Edit" button on the menu, WFE identifies the selected text from the HTML text and dynamically generates a form for editing on the Web page (Figure1-(b)). As shown in Figure 1-(b), the chosen text is in the form. If the text is decorated with HTML tags, the tags are also in the form. The user edits the text in the form and then clicks the "Save" button to save the edited text. In this example, the user inputs the HTML tag

```
    <font color="red" size="+1">
```

before "November 8, 2004". That is, the user tries to modify the color and size of the text.

Figure 2 shows another example in which the user wants to attach a note to the paper submission information. The user clicks the right mouse button at the position where he/she wants to attach a note, and WFE opens an appropriate menu in Figure1-(a). From the menu, the user can select the type of note, i.e., the "Comment" note or the "Image" note (Figure 2-(a)). If the user selects the


WFE -Paste Menu-
Comment
Image
Cancel
"Comment" is the function that an user can attach a text anywhere
"Image" is the function that an user can attach an image anywhere
(a)
<font color="red"> I must submit
 paper by November 5, 2004 </font>
Save
The "Comment" function opens TextArea to input text
(b)
Select Image File...
alert.gif
Paste
Cancel
Menu to select an image file. In this figure, "alert.gif" is selected
(c)
(d)

WWW2005 Important Dates

http://www2005.org/date.html

Top ▸ Important Dates

Important Dates

I must submit
my paper by November 5, 2004

	DEADLINE	NOTIFICATION
Paper submission:	November 8, 2004	January 28, 2005
Final papers due:	February 28, 2005	
Poster submission:	February 7, 2005	March 7, 2005
Industrial track submission:	February 7, 2005	March 15, 2005
Tutorial/workshop proposals:	October 15, 2004	November 15, 2004
Panel proposals:	November 8, 2004	January 31, 2005
Conference:	May 10-14, 2005	

Author notification dates have been changed from the originalannouncement. Paper author notification changed from Jaunuary 24to Janunary 28 and poster one changed from February 14 to March 7.Tutorial/workshop proposals deadline changed from October 11 toOctober 15.

Paper submissions must be received by *11:59pm Hawaii Standard Time* on November 8th.
There are no extensions allowed.

Contents Menu
- Top
  - CFP
  - Important Dates
  - Schedule
  - Chairs
  - Mailing List
  - News
  - Sponsors / Partners
- Papers
- Tutorials / Workshops
- Panels

Fig. 2. Attaching notes using WFE

"Comment" button, WFE opens a text area form in which the user inputs a comment note, as shown in Figure 2-(b). If the "Image" button is selected, WFE opens a menu for selecting an image file (in Figure 2-(c)). Figure 2-(d) shows a page on which a comment and an image have been attached. The user can move/delete/edit the attached comment and image by using the menu that appears with a right mouse click at their position. The edited data is stored and maintained by the proxy agent. While users who browse the page using the proxy agent view the edited page, other users view only the original page. Accordingly, by using a private proxy agent, the user can keep the edited data. WFE not only enables users to pseudo-edit all Web pages, but also enables users to generate new Web pages using only their Web browser without suffering from trouble with HTML.

3. The architecture of WFE

Figure 3 shows the architecture of WFE, which is based on a proxy agent and a script for dynamic editing function based on JavaScript. The proxy agent is implemented using MiLog [5], which is an agent development framework based on Java. Since the MiLog agent has a WWW server function and a Web proxy function, we can easily implement an agent for realizing a Web proxy. Moreover, we can program a MiLog agent based on a logic programming language like Prolog. Thus, a pattern-matching mechanism for identifying selected a portion of text from an HTML text can be effectively realized. The proxy agent for WFE has three main functions as follows:

- Caching HTML files downloaded from target URL (in Figure 3-i): When a user sends a request for a certain URL via the proxy agent, the agent caches the returned HTML files in a database (a working memory) of the agent.
- Attaching JavaScript for pseudo-editing functions to each cached HTML file and send the files to a Web browser (in Figure 3-ii): The proxy agent attaches JavaScript to each HTML file to enable pseudo-editing using a Web browser. Since the proxy agent already has a function for a proxy server, there is no need to

implement a mechanism for sending HTML files to the browser. When a user access a web page via the proxy agent, the proxy agent augments the web page with client-side scripting via JavaScript that provides the user with the editing functions.

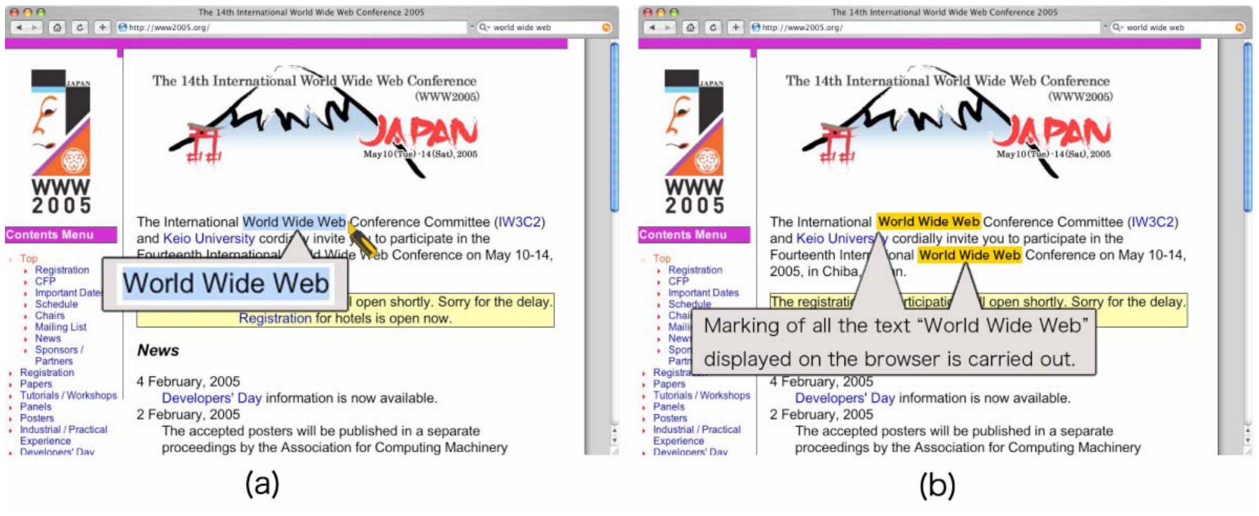

World Wide Web
Marking of all the text "World Wide Web" displayed on the browser is carried out.
(a)
(b)

Fig. 5. Execution example of WPM

- Modifying cached HTML files based on edited data (in Figure 3-iv): When a user edits an HTML file in a Web browser, the script added to the HTML file sends the edited data in the file to a proxy agent. The proxy agent then modifies the cached HTML file using the edited data and stores it in the database.

The pseudo-editing function in WFE is based on JavaScript and DOM. WFE can handle mouse actions and send edited data to a proxy agent. The outline of the mechanisms of the functions can be described as follows:

- Receiving an HTML text selected by a user and identifying it in HTML file (in Figure 3-iii): Since the HTML tags decorating the selected text must be extracted with the selected text, the pseudo-editing function must identify the selected text in the HTML file and retrieve the tags around it.

- Generating a form for editing (in Figure 3-iii): The dynamic editing function generates a form that includes the text and tags identified in Figure 3-iii.

- Rendering only the edited part of HTML text (in Figure 3-iii): The dynamic editing function replaces only edited text, so bothersome reloads of Web page are avoided.

- Sending edited data to the proxy agent (in Figure 3-iv): We explain the detail of the proxy agent and added JavaScript codes in the remaining sections.

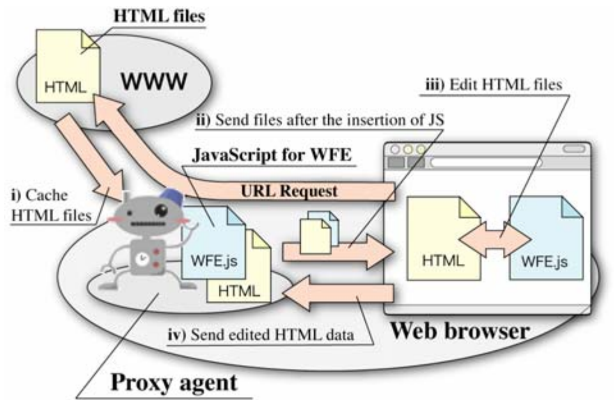


Fig. 3. The architecture of WFE

3.1 The proxy agents in WFE

A proxy agent is allocated to each user and manages the pseudo-editing functions on the user's computer. A user simply specifies the address of his/her proxy agent for proxy setting in a Web browser. The proxy agent analyses an HTTP header sent from the Web browser to get the URL. The agent then downloads and caches the HTML files for the URL. Next, the agent adds JavaScript to the HEAD tag of each file. In practice, the added JavaScript indicates a JavaScript file to use for dynamic editing function. If there is no HEAD tag, the agent automatically adds one. A sample of the script added to the HTML file is as follows:

```
<script type="text/javascript">
```

```
    var TargetID = (ID of Web page)
    var AgentAddress = (Address of proxy agent)
  </script>
  <script type="text/javascript"
    src="(Address of proxy agent)/wfe.js">
  </script>
```

The `TargetID` is an unique ID for each HTML file. The proxy agent uses this ID to specify the cached HTML file to be modified. The `AgentAddress` is the IP address with a port number of the proxy agent. If the proxy agent works on a Web server, the IP address of the server is used for the `AgentAddress`. The ``wfe.js" is the JavaScript file for dynamic editing function, which is explained in the next section.

3.2 The framework for dynamic editing

When a user selects a target text on the Web browser by dragging the mouse cursor, the Dynamic Editing Function (DEF) gets the selected text, and searches for it from HTML file by pattern-matching. The process is then branched out according to the number of texts that match with the selected one. If there is only one matching text, the DEF identifies the target text to be edited by the user. On the other hand, if there are several matching texts, the DEF must identify the position of the text in the HTML file. For this identification, we use a SPAN tag. That is, the DEF inserts a SPAN tag with an original name attribute, and in conjunction with the insertion, the DEF gets the mouse cursor position when the user selects the text. Next, the DEF gets the position of each SPAN tag (precisely, each text between `<SPAN>` and `</SPAN>`. The text nearest to the mouse cursor position is identified as the one to be edited by the user. After identifying the text, the DEF replaces the selected text with the text-field form into which the selected text is inputted. When the user has finished editing, the DEF replaces the text field form with the inputted text. If the selected text is wedged by SPAN tag, the DEF deletes all SPAN tags in the HTML file.

Figure 4 shows the pseudo script that is a part of the script for the DEF. We suppose this script processes an HTML file in which there are multiple texts that match with the selected text. In the first two lines in Figure 4, the selected text and the text wedged by BODY tag are substituted into variables, respectively. In the lines 3 and 4, the DEF inserts SPAN tag. To put it concretely, the selected text “selectedTxt” is replaced with “`<SPAN>selectedTxt</SPAN>`”. Even if there are multiple matching texts, all those texts can be replaced because the replacing process is recursively executed. In the line 5, the DEF gets a list of SPAN tags, and then in the line 6, the nearest SPAN tag from the mouse cursor position is returned by the original function “getNearestNode()”. Finally, in the lines 7 and 8, the nearest SPAN tag and the wedged text is replaced with the text field form. The function “submitText()” in the line 7 deletes all SPAN tags.

Pseudo Script: For Text Replacing

```
1: var selectedTxt =
      document.getSelection();
2: var htmlSource =
      document.getElementsByTagName
         ("body")[0];
3: var reg = new RegExp(selectedTxt, "g");
4: var htmlSource = htmlSource.replace(reg,
      "<SPAN name='wfe'>" + selectedTxt +
      "</SPAN>");
5: var aList = htmlSource.getElementsByName
      ("wfe");
6: var index = getNearestNode
      (Mouse.x,Mouse.y,aList);
7: var formHTML = '<INPUT name="txtfield"
      type="text" size="18" value="'+
      selectedTxt + '"><INPUT type="submit"
      value="Save" onclick=
      "submitText();">';
 8: aList[index].innerHTML = formHTML;
```

Fig. 4. A pseudo script to replace selected text

To sum up, the DEF can precisely identify the selected text without complicated text processing. Accordingly, the DEF exerts a light load on the computer.

4. The Web Page Maker

WPM (Web Page Maker) is a Web browsing support system implemented based on WFE, which makes it possible to carry out marking to existing Web pages by operation on a web browser. WPM can improve the browsing experience of the existing browser. By choosing a text with a mouse cursor, a font of the selected portion, a background color, a character color, character size, etc. can be changed. Since the operation of the system is performed by the operation similar to carrying out marking to paper, it is intelligible for a user. WPM has also an anchor creation function to a Web page that carried out marking. By enabling creation of anchors in arbitrary parts, the ease of reading the web page is improved and revisit is supported.

Figure 5 shows an example of marking using WPM. First, a user selects the target text ``World Wide Web", by mouse dragging in the Web browser (in Figure 5-(a)). Next, he/she clicks the right mouse button to carry on marking (in Figure 5- (b)). When the user does marking to some particular text on the Web page, the same text else where on the same page is automatically marked. The user can grasp visually where the word that carried out marking is contained.

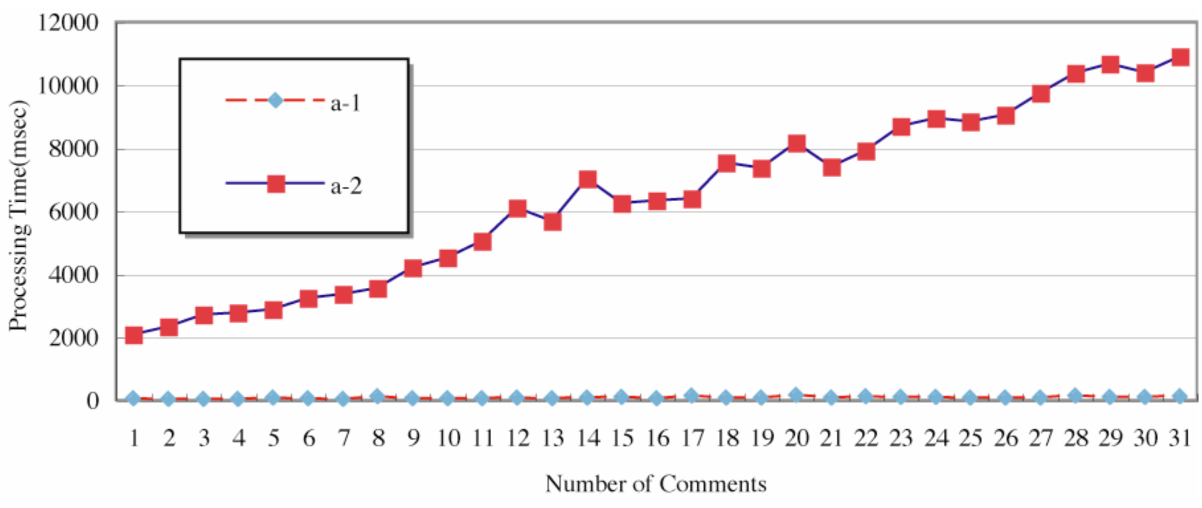


Fig. 6. The experimental result of WFE

The system architecture of WPM is based on a proxy agent and marking mechanism implemented by using WFE. The proxy agent adds marking mechanism to the HEAD tag of each HTML file. A user activates a marking mechanism on a Web browser by using a mouse action. When a text is chosen on a Web browser, the marking mechanism will change the character decoration of the text on the browser. The information used for marking is stored in a database of the proxy agent.

5. Discussion

When a user edits text using WFE, he/she does not need to reload the Web page. WFE can display the edited page instantaneously. This is because WFE does not render the whole page. WFE simply replaces the edited text with JavaScript. We call the editing the pseudo-editing. Figure 6 shows the experimental result of the process of adding comments to a Web page by using WFE. The vertical axis shows the processing time (msec). The horizontal axis shows the number of the comments. The size of the original page is 8,509 bytes, and the size of the comments is between from 90 to 110 bytes. In the Figure, the a-1 plots the graph for the processing time of the pseudo-editing in which results are reflected when a user add comments to a Web page by using a cache memory of a proxy agent. The a-2 shows the processing time in which a proxy agent merges an original Web page and the added comments to generate a new Web page on a Web server. The result shows the pseudo-editing of WFE can be effectively used for large scale Web applications.

WFE users can share useful annotations within a community while the privacy of the information is ensured [11]. Users in a community share annotations via their proxy agents. Because, other users do not normally know about the proxy agents, they cannot view the shared annotations. If a user actually wants to make an annotated Web page public, he/she can do so by generating a new Web page and storing it on a Web server. The proxy agent generates the new page by merging the original page with the annotations attached to the page. There are several annotation systems. WBI [1] and Crit-Link [3] are existing annotation system which has similar functions to WFE. Especially, WBI has the similar function to WFE. WBI stands between an user and the Web for monitoring, editing, and generating documents. Additionally, WBI introduces several agents to observe each user's action. However, these systems do not allow users to freely and flexibly edit common Web pages. For example, users cannot edit documents on a Web page and attach texts and images on the arbitrary place on the Web page. On the other hand, in WFE, users can freely edit the Web pages.

6. Conclusions

We have described the WFE (Web Flexible Editing) method, for online editing of Web pages. WFE users can pseudo-edit common Web pages on the Internet without needing to know particular scripts and HTML tags. Furthermore, since the result of editing is instantaneously reflected, users do not need to reload the page to see the results. We are currently trying to apply the WFE method to develop online education tools [2] as Web applications. In Our approach, schoolchildren in the lower grades can fully utilize the Web tools [10]. The evaluation of the Web tools is an ongoing work [8]

We also described WPM, a Web browsing support system to which WFE is applied. WPM enables a user to do marking on a Web page like marking on paper. Advantages of WPM can be described as follows; (1) In arbitrary Web pages, marking is made possible, (2) WPM can use from existing common web browsers and there is no necessity that a user prepares plug-in and special software component.

The result shows that the method WFE is effective in developing Web applications such as the system WPM.

References

[1] Barrett, R., Maglio, P. P., and Kellem, D. C. 1997. How to personalize the web. In In the Proc. of the ACM Conference on Human Factors in Computing Systems (CHI-97), pages 75-82.
[2] Chiu, B. C., and Yu, T.T. 2002. Promoting the use of information technology in education via lightweight authoring tools. In In the Proc. of the International Conference on Computers in Education, pages 501-505.
[3] Yee,K. 2002. Critlink: Advanced hyperlinks enable public annotation on the web. In ACM Conference on Computer Supported Cooperative Work, (Demonstration Abstract).
[4] Firefox, http://www.mozilla.org/products
[5] Fukuta, N., Ito, T., and Shintani, T. 2001. A logic-based framework for mobile intelligent information agents. In In the Proc. of the 10th International World Wide Web Conference (WWW10), pages 58-59.
[6] Google Toolbar, http://toolbar.google.com
[7] Leuf, B., and Conningham, W. 2001. The Wiki Way: Quick Collaboration on the Web, Addison-Wesley.
[8] Matsuo, T., Ito, T., and Shintani, T. 2004. Qualitative/Quantitative Methods-Based e-Learning Support System in Economic Education, Proc. of the 19th National Conference on Artificial Intelligence (AAAI-2004), pp592-598.
[9] Obendorf, H., and Weinreich, H. 2003. Comparing link marker visualization techniques: changes in reading behavior, Proceedings of the twelfth international conference on World Wide Webpp. 736-745.
[10] Tanaka, M., Matsuo, T., Tashiro, N., Nishi, K., Ito, T., and Shintani, T. 2004. An implementation of web-based interactive integrative learning supporting system. In In the Proc. of the 2004 International Conference on Artificial Intelligence (IC-AI'04).
[11] Tashiro, N., Hattori, H., Ito, T., and Shintani. T. 2004. Implementing a proxy agent based writable web for a dynamic information sharing system. In In the Proc. Of the 13th World Wide Web Conference (WWW-2004), pages 256-257.
[12] Wiki wiki web, http://c2.com/cgi/wiki

Tadachika Ozono received his bachelor degree in engineering from Nagoya Institute of Technology, his master degree in engineering from Nagoya Institute of Technology, and his Ph.D in engineering from Nagoya Institute of Technology. He is a research associate of the Graduate School of Computer Science and Engineering at Nagoya Institute of Technology. His research topic is a web intelligence using multiagent and machine learning technologies.