

Improving Accessibility for Existing Websites Spanning Multiple Domains

Takashi Sakairi¹, Takuya Ohko², and Hidemasa Muta¹

¹ IBM Japan, Tokyo Research Laboratory

² IBM Japan, Accessibility Center

1623-14, Shimo-tsuruma, Yamato, Kanagawa 242-8502, Japan

{sakairi, ohkot, hmuta}@jp.ibm.com

Abstract. There are demands for improving accessibility in existing websites by enlarging text and changing the colors. Typical solutions use technologies such as ActiveX that can only run on a specific client environment. JavaScript is supported in many Web browsers, and it can be used to add new functions for improving the accessibility of existing websites. However, Web browsers prohibit JavaScript from accessing webpages of another domain, so it is difficult to improve accessibility for related websites spanning multiple domains. This paper describes a method that solves the problem.

Keywords: Accessibility, Websites, Dynamic HTML, JavaScript, Same Origin Policy.

1 Introduction

The Web plays important roles in daily life and business. Many organizations develop websites for various purposes, such as to deliver information and for electronic commerce. Meanwhile, the ratio of senior citizens is increasing and will soon be increasing quite rapidly in many countries. However, there are many obstructions to senior citizens' access to the Web. Due to the complications of weak eyesight, many seniors are uncomfortable about browsing webpages crammed full of information and using small, hard-to-read fonts. In response, many websites' owners recognize that accessibility is increasingly important [1, 2, 3]. It is possible to improve accessibility for people with limited vision by enlarging the text and by changing the colors.

Fig. 1 shows two types of accessibility tools. The first type is an accessibility tool for a Web browser (a). Users install such an accessibility tool by themselves [4]. Users can access any Web server with the tool, but it is difficult for novice users to install such tools. The second type is an accessibility tool for a Web server (b). Providing accessible websites without troubling the user is important for many organizations, especially for public institutions. This paper describes an accessibility tool for a Web server.

When a website's owner decides to improve accessibility, it is not always the case that the affected webpages are in a single domain. For example, a local government may have a website, <http://www.city.jp/>, but a university established by the local government may have a separate website, <http://www.u-city.ac.jp/>, in another domain.

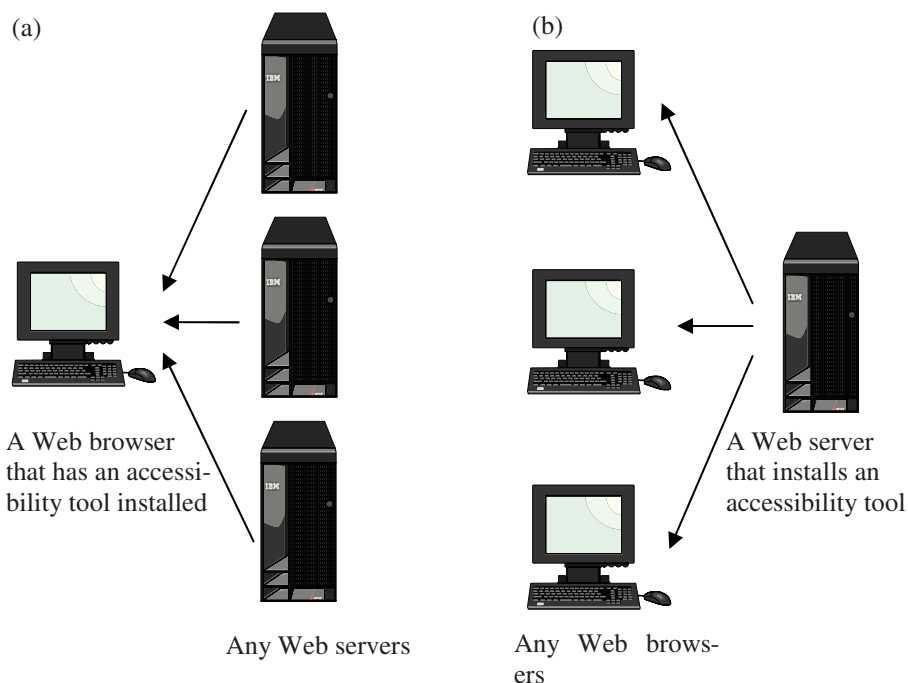


Fig. 1. Two types of accessibility tools: (a) an accessibility tool for a Web browser, and (b) an accessibility tool for a Web server

Fig. 2 shows three methods for a website's owner to improve the accessibility of the website. The first method is by generating personalized webpages on the Web server (a). The text size, the text color, and the background color of the webpages are changed according to each user's preferences. This method is not acceptable for many website owners, because they already have a lot of content in their website and this method usually requires modification of the existing content to parameterize it. The second method is transcoding webpages at a Web intermediary [5, 6, 7] (b). This method does not require modification of the existing content, but lots of computing power is required for the Web intermediary if the number of users is large. The third method involves sending a program to the Web browsers, and the program modifies the appearance of the webpages on the Web Browsers [8, 9, 10] (c). This method also requires no modification of the existing content. In addition, this method is scalable, because the changes of the webpages' appearances are made by the users' machines. This paper describes an approach using the third method.

Most of the proposed solutions [8, 9, 10] to improve websites' accessibility use technologies available only on a specific client environment (such as ActiveX for Internet Explorer). With such solutions, users have to use a specific operating system and a specific Web browser. Our work addresses this problem.

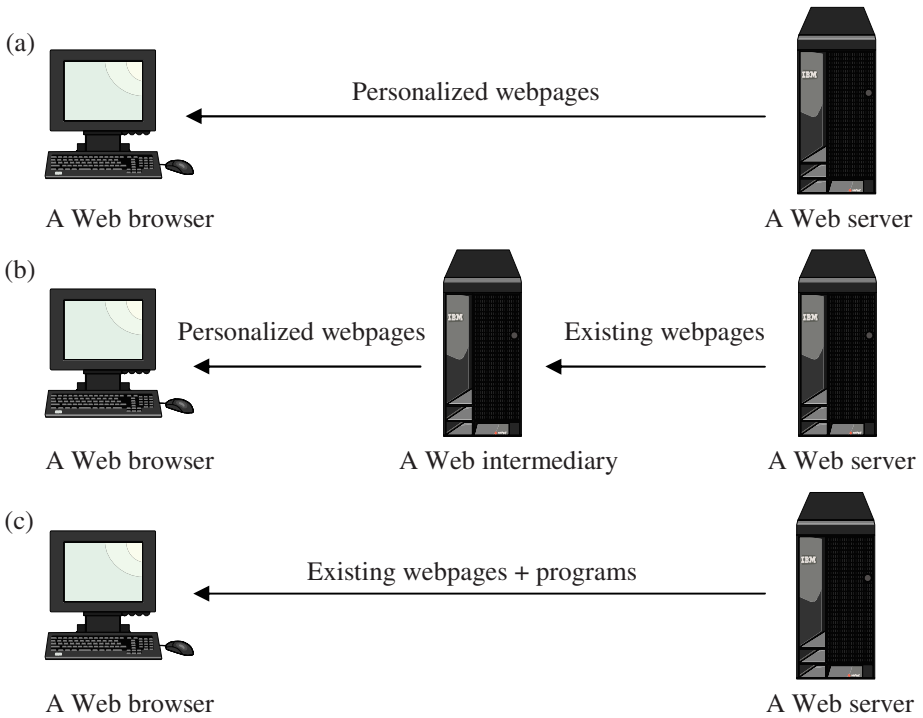


Fig. 2. Three types of method to improve website accessibility: (a) generating personalized webpages on a Web server, (b) transcoding webpages by a Web intermediary, and (c) sending a program to a Web browser

The remainder of this paper is organized in the following way: Section 2 presents the basic methods to improve accessibility for an existing website using JavaScript™. Section 3 suggests ways of enhancing the method for related websites in multiple domains. Section 4 concludes the paper and describes our future work.

2 Improving Accessibility Using JavaScript

Our proposed method uses JavaScript [11] to add new functions to improve accessibility. JavaScript is supported in many Web browsers and such Web browsers run on many operating systems, so users can access these accessible websites from heterogeneous client environments. In addition, users do not have to install a program when using JavaScript. JavaScript programs are executed without special installation.

A webpage loaded in a Web browser is represented by a tree structure called a Document Object Model (DOM) [12]. JavaScript can handle events generated by a DOM, and can modify the DOM to change the webpage's appearance. For example, JavaScript can highlight a text element under the mouse pointer, and can show the text in another frame with larger text. Fig. 3 shows an example of enlarged text.

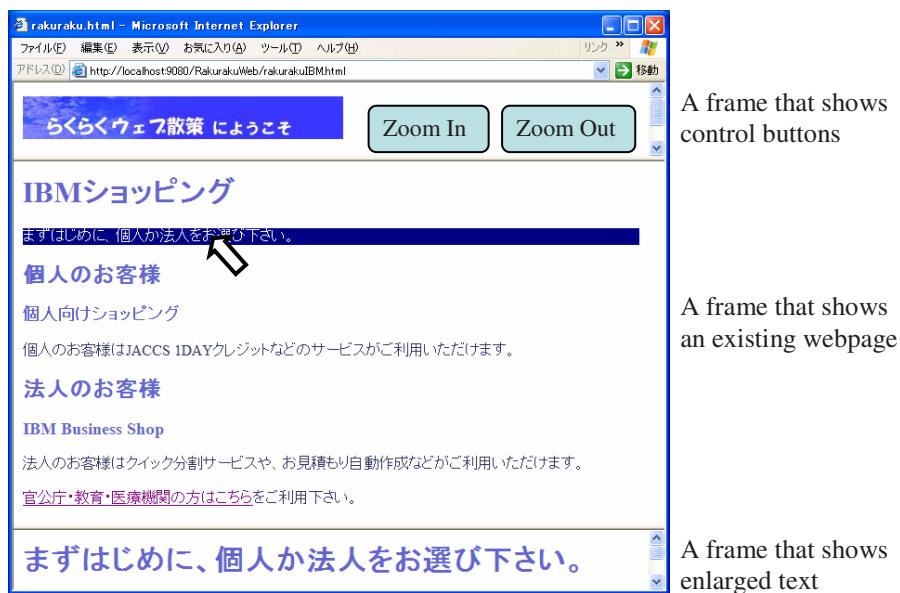


Fig. 3. Example of enlarged text

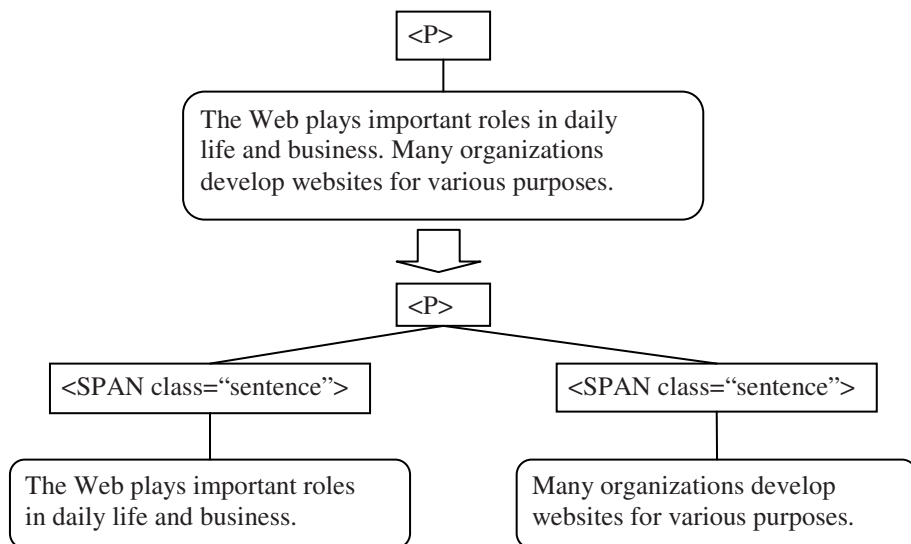


Fig. 4. Modification of a DOM

Our method uses several frames: one frame (the content frame) is used for displaying existing webpages and the other frames are used for additional functions such as displaying buttons and showing enlarged text. When an existing webpage is loaded in the content frame, an “onload” event handler for the content frame is called. (Although the

“onload” event handler for the “FRAME” element is not actually defined in HTML specification [13], it is enhanced in both of the major Web browsers Internet Explorer and Mozilla.) The handler modifies the appearance of the webpage and sets event handlers for “mouseover” and “mouseout” events on the page. By setting up the event handlers, we can manipulate the webpage’s DOM when events are generated.

The “mouseover” event occurs when the pointing device is moved onto an HTML element, and the “mouseout” event occurs when the pointing device is moved so it is no longer over an HTML element. We can highlight the pointing HTML element, and show the enlarged text in the other frame by the event handlers.

It is sometimes necessary to modify the DOM in the “onload” event handler to access fine-grained mouse events. For example, a “P” element can be divided into several “SPAN” elements to get mouse events for the individual sentences (Fig. 4).

When a link in a webpage (in the content frame) is followed to another webpage in the same domain, the “onload” event handler is executed and the accessibility functions work fine. However if it is followed to another webpage in another domain, the “onload” event handler cannot be executed and the accessibility functions do not work. Script programs cannot access a DOM loaded from another domain. This is called the Same Origin Policy [14].

3 Enhancement for Multiple Domains

We enhanced our method so that the added functions for improving accessibility can be used without requiring the users to make special efforts between related websites

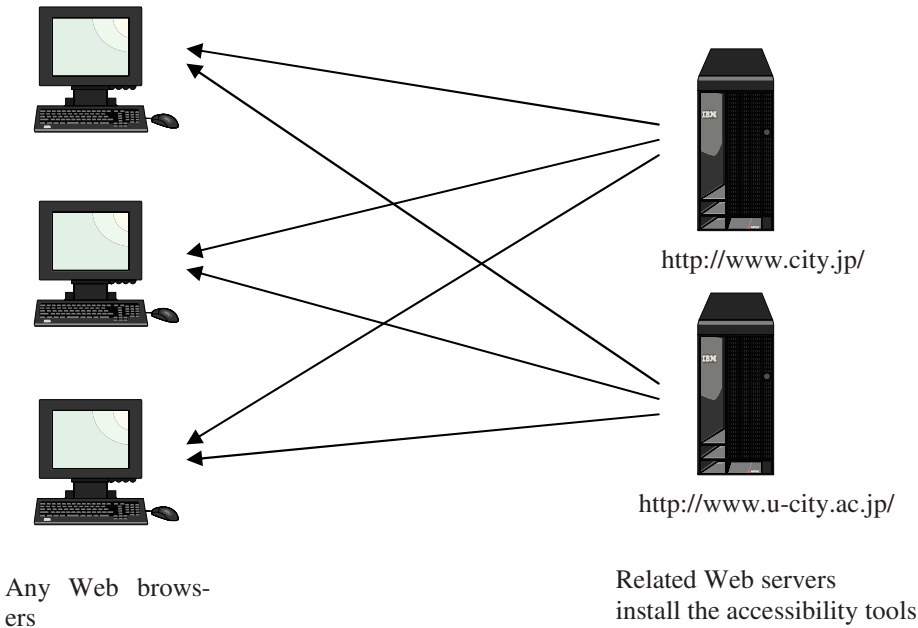


Fig. 5. Related Web servers install the accessibility tools

in multiple domains. Suppose a website's owner want to improve the accessibility for two related websites such as <http://www.city.jp/> and <http://www.u-city.ac.jp/> (Fig. 5). And some of the webpages in the websites are linked to each other. Our goal is to let users navigate from one website to the related website while using the accessibility capabilities without special efforts.

Each website has a dynamic webpage for generating an accessible page with a frameset: <http://www.city.jp/EasyWeb> and <http://www.u-city.ac.jp/EasyWeb>. The URL of the content frame is specified as a parameter. It is possible to access the webpage <http://www.city.jp/news.html> with improved accessibility by specifying <http://www.city.jp/EasyWeb?content=news.html>.

Following is an example of generating an accessible webpage implemented with JavaServer Pages™. The second frame is a content frame, and the “src” attribute is specified by the “content” parameter. The other frames are used for additional functions.

```
<HTML>
<HEAD>
<%@ page language="java" contentType="text/html" %>
<TITLE>Easy Web</TITLE>
<SCRIPT type="text/javascript" src="easy.js"></SCRIPT>
</HEAD>
<FRAMESET rows="20%,*,30%">
  <FRAME src="controller.html" name="controllerFrame">
  <FRAME src="<%=request.getParameter("content")%>"
    name="contentFrame" onload="init(contentFrame)">
  <FRAME src="magnifier.html" name="magnifierFrame">
</FRAMESET>
</HTML>
```

We added the following two steps to the “onload” event handler for the content frame to support multiple domains. First, we check the “href” attribute of all “A” elements and the “action” attribute of all “FORM” elements in the content frame’s page. Next, if the attribute value is specifying a page in a related website, we change the value to be a corresponding URL for improved accessibility, and change the “target” attribute to “_top”. “_top” means that loading a new webpage does not occur into the content frame but into the topmost window. When the user navigates to the modified URL, the webpage contains a “FRAMESET” and the webpage in the content frame have the same origin so the “onload” event handler can execute. By using these steps, users can navigate to the pages of the related websites with improved accessibility.

4 Conclusion and Future Work

We developed a prototype system for our proposed method, and confirmed that our method can improve accessibility for existing websites spanning multiple domains. We also confirmed that our method works on several combinations of operating

system and Web browser: Windows™ with Internet Explorer, Windows with Mozilla, Linux™ with Mozilla, and Mac with Mozilla.

The IBM® Tokyo Research Laboratory developed an assistive Web browsing system, Easy Web Browsing [8, 9], in 2002. The system has already been adopted by some IT leaders among local governments, ministries of the central government, and IT companies in Japan and other countries. It enlarges the displayed characters, changes background and character colors, and reads text aloud. The currently released version is implemented with ActiveX, so it works only on the combination of Windows with Internet Explorer. Although installing Easy Web Browsing is not difficult, users do have to install it on their computers.

We plan to enhance Easy Web Browsing by using our new method. Although it is possible to implement most functions required for improving the accessibility of web-pages by using JavaScript, it is not possible to implement some functions that way. For example, reading text aloud is not possible with JavaScript. We have to implement such functions by using another technology, and call the functions from JavaScript.

References

1. World Wide Web Consortium, Web Accessibility Initiative (WAI) <http://www.w3.org/WAI/>
2. Section 508: The Road to Accessibility <http://www.section508.gov/>
3. Rowan, M., Gregor, P., Sloan, D., Booth, P.: Evaluating Web Resources for Disability Access. In: Proceedings of the 4th ACM Conference on Assistive Technologies, pp. 80–84 (2000)
4. Asakawa, C., Itoh, T.: User Interface of a Home Page Reader. In: Proceedings of the 3rd ACM Conference on Assistive Technologies, pp. 149–156 (1998)
5. Barrett, R., Maglio, P.P.: Intermediaries: An approach to manipulating information streams. IBM System Journal 38(4), 629–641 (1999)
6. Huang, A.W., Sundaresan, N.: A Semantic Transcoding System to Adapt Web Services for Users with Disabilities. In: Proceedings of the 4th ACM Conference on Assistive Technologies, pp. 156–163 (2000)
7. Fairweather, P.G., Hanson, V.L., Detweiler, S.R., Schwerdtfeger, R.S.: From Assistive Technology to a Web Accessibility Service. In: Proceedings of the 5th ACM Conference on Assistive Technologies, pp. 4–8 (2002)
8. Muta, H., Ohko, T., Yoshinaga, H.: An Active-X-based Accessibility Solution For Senior Citizens. In: Proceedings of CSUS's 20th Annual International Conference (2000)
9. IBM Japan, Easy Web Browsing http://www.ibm.com/able/solution_offerings/EasyWebBrowsing.html
10. Hitachi Government & Public Corporation System Engineering, ZoomSight http://www.zoomsight.jp/demo/zsmd/lang/en/html/help/about_help.html
11. ECMA International, ECMAScript Language Specification, Standard ECMA-262, 3rd edn. (December 1999) <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
12. World Wide Web Consortium, Document Object Model (DOM) Level 2 HTML Specification Version 1.0, W3C Recommendation (January 09, 2003) <http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109/>

13. World Wide Web Consortium, HTML 4.01 Specification, W3C Recommendation (December 24, 1999) <http://www.w3.org/TR/1999/REC-html401-19991224/>
14. Ruderman, J.: The Same Origin Policy
<http://www.mozilla.org/projects/security/components/same-origin.html>

IBM is a registered trademark of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.