Proposal for Solving Incompatibility Problems between Open-Source and Proprietary Web Browsers

Jun Iio, Hiroyuki Shimizu, Hisayoshi Sasaki, Akihiro Matsumoto

Proposal for Solving Incompatibility Problems between Open-Source and Proprietary Web Browsers

Jun Iio[1], Hiroyuki Shimizu[1], Hisayoshi Sasaki[2], and Akihiro Matsumoto[2]

[1] Research Center for Information Technology, Mitsubishi Research Institute, Inc.
{iiojun,hshimizu}@mri.co.jp
[2] Gluegent, Inc. {sasaki,matsumoto}@gluegent.com

Abstract. We conducted demonstration experiments to promote the use of open-source software on desktops as a part of the Open-Source Software-Utilization Development Program for the education sector. In these experiments, we found some barriers to popularizing the use of open-source software in end-user desktop applications. In this paper, we report some typical problems of Web-browser compatibility, which are considered to be obstacles for promoting open-source software on desktops. We also introduce a tool that we developed to help developers avoid such pitfalls while designing Web applications.

1 Introduction

In the late 1990s when open-source software was not considered as a commodity, open-source server software was used much more widely than open-source desktop applications. Recently, the open-source desktop applications have also become popular, and some well-known applications such as OpenOffice.org, Firefox, and Thunderbird are gaining market share and awareness on the Internet. According to the report from the open source application foundation (OSAF)[1], the use of Linux on desktops will first become popular among high-end users and IT engineers, then its use by routine workers (such as call-center operators) will follow. Ultimately, even office workers will use open-source software on their desktops.

2 Background

In order to accelerate the spread of open-source applications on desktops, we conducted a series of demonstration experiments from 2004 to 2008 as part of the Open-Source Software-Utilization Development Program for the education sector, which was sponsored by the Information-Technology Promotion Agency Japan (IPA) and the Center for Educational Computing (CEC). In these experiments, Linux desktop computers were deployed at more than 50 schools, and students used Linux every day in the classroom. As a results of these experiments, we demonstrated that there were almost no problems using open-source

desktop computers instead of the existing proprietary computers for IT-driven classes. In addition, we showed the possibility of reducing management cost of IT systems by adopting open-source software.

On the other hand, our study based on those experiments revealed many problems that must be solved in order to replace proprietary software with open-source software on desktop computers. One of the key issues in the use of open-source applications in the field of education is the ability to handle established course material for e-learning systems.

Many e-learning systems are constructed as Web applications. In general, Web browsers are used as client software, and viewing course material via the Internet requires a Web browser. However, much educational content is designed to be viewed by a particular Web browser (many e-learning systems are built to run on Microsoft Internet Explorer (MSIE) and they are tested only on that browser), In addition, many Web pages are consistently produced with defective content that cannot be properly viewed by open-source browsers.

Obviously, the trend toward such inappropriate Web content exists not just in the e-learning field, but also in general Web-based applications in other fields. Therefore, a study to identify browser-incompatibility trends was required; it was essential to clarify the practical problems and solutions in order to popularize the use of open-source browsers.

3 Browser Dependency Problems

Following are the typical types of browser-incompatibility problems.

Embedding of Video and Other Objects. This problem is caused by a difference in the handling of two tags: `<embed>` and `<object>`. To create a document that contains such an object and that is rendered correctly in every browser, the object must be embedded in the document using both tags.

Requirements of Particular Plug-ins. Shockwave, XVL Player / Viewer, and certain other plug-ins using Active-X technology cannot be handled in the Linux.

Unsupported Functions in Japanese documents. Text is written in two directions. MSIE implements the product-specific extension "writing-mode: tb-rl" in its style-sheet definition to enable vertical writing mode. However, Mozilla Firefox does not support this function.

Layout Fragmentation by Line-Spacing Misalignment. Some documents specify line spacing using a property of the tag `<p>`. This is valid only in MSIE, and not valid in Firefox. According to HTML specification, using this tag property is inappropriate, and the `<div>` tag should be used to adjust line spacing instead.

Garbled Characters. In the representation of Japanese text, there are four sets of character codes. Especially in Java applications, not assigning explicit character codes for the text can cause garbled characters to occur.

Difference in the Versions of Java Runtime Environment (JRE). Some systems cannot run as expected because of issues with Java applet initialization.

Defects in Different JavaScript (ECMAscript) Implementations. Because of differences in the implementation of JavaScript in MSIE and Firefox, many malfunctioning scripts, which have been tested only on MSIE, have been encountered.

3.1 Solutions for This Problem

These problems are serious, because Internet has achieved an important position in the infrastructure of the information society.

We investigated the problems that exist in the Internet space and their effect on Internet users. The result of our investigations [2] revealed that approximately 170 incompatibility problems potentially exist in the Internet. Some of other problems were categorized as fatal errors. For example, some data would never get displayed in a particular browser. Based on our findings, a recommendation document [3] was published.

As a solution to this difficult situation, we designed a tool for Web designers, Web developers, and Web application programmers to make the development of multi-browser Web application quick and easy. The key advantage of this tool is the information it offers to Web developers to avoid issues affecting interoperability. The "Pirka'r" system, which is introduced in this paper, was developed and released to the Internet as an open-source software. Pirka'r version 1.0 was released at the end of September 2009. It is now available under the Apache License, Version 2.0, from `http://pirkar.ashikunep.org/`.

4 System Overview

In this section, we provide an overview of Pirka'r and its ecosystem.

4.1 Overview of Pirka'r

Pirka'r is an integrated development environment for Web designers and Web application developers consists of two parts: a client running on the user's workstation, and a verification server. The verification server can be installed either on the workstation or on a separate PC.

The Pirka'r front-end was developed as an application in Eclipse Rich Client Platform (RCP). Following are the main functions available in Pirka'r:

- Verification against standards.
- Assessment of multi-browser interoperability.
- Multi-browser previewing.
- Automatic rendering.
- Multi-functional editing.

– Auto-downloading of a set of Web pages.

The Pirka'r user can verify whether the cascading stylesheet (CSS) definition in his Web content complies with standards. In addition, HTML/CSS/JavaScript can be examined in regard to browser-dependent descriptions.

The target Web page can be displayed in multiple browser panes. This function relieves the designers of the annoyance of handling multiple browsers and checking rendering results in each browser.

Web content written by the multi-functional editor is rendered in the multi-browser preview panes in real time. Designers are freed from the need to reload manually. The multi-functional editor is a HTML/CSS/JavaScript editor, which has convenient functions such as code completion, syntax highlighting, and grammatical checking.

4.2 The Pirka'r Ecosystem

Figure 1 shows an overview of the Pirka'r ecosystem.

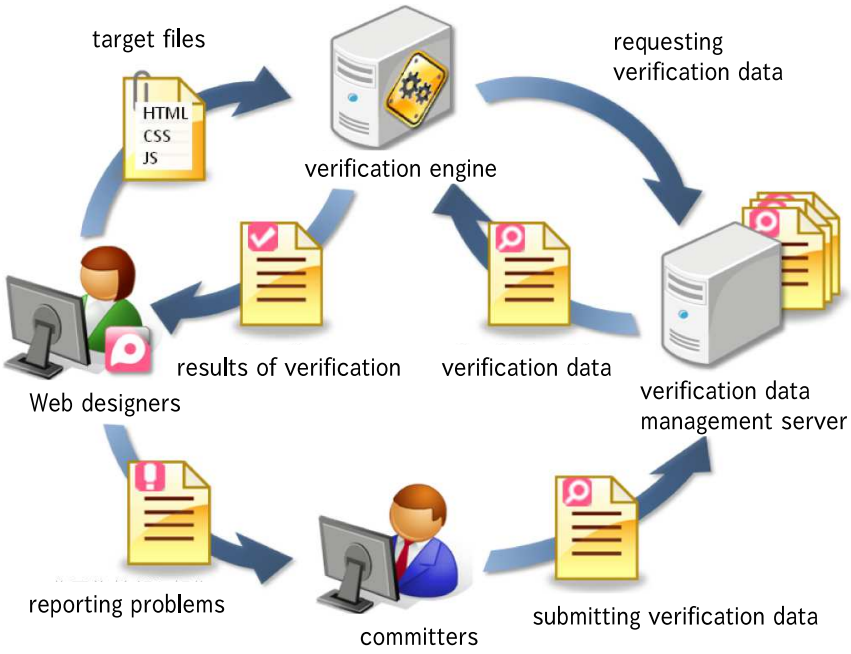

Fig. 1. Overview of the Pirka'r ecosystem.

The function that verifies interoperability problems on target Web content uses a verification engine running separately with the Pirka'r client. The verification engine provides the results of verification processes, which are based on verification data provided from the verification data-management server.

A single management server that manages verification data can deliver the set of verification data to the verification engines that are installed locally on any network.

If a designer detects a new interoperability-discrepancy problem, he can report the problem via the reporting form provided. When we receive the report, committers in our community attempt to create new verification data that contains a check script, as well as a series of documents that show how to fix the problem.

4.3 Verification Script

The verification script is written in JavaScript. It searches for defects in the target Web content, traversing over its Document Object Model (DOM).

If the problem is not too complex, the verification script is quite simple. We have already prepared more than 100 scripts to check existing problems, and stored them in the master database. We are now asking committers who belong to the Seaser Foundation to update verification scripts. Seaser Foundation is one of the most famous organizations in Japan associated with Web application development.

Committers can register new verification data using the submission form of the verification data-management server, which is also managed by the Seaser Foundation. A set of verification data consists of a name, category (HTML, CSS, or JavaScript), language, severity level, reason for the problem, proposals to solve this problem, verification script, examples of descriptions, and attached files (if any).

5 Related Works

A study on interoperability-discrepancy problems associated with Web standards was also carried out by the standard and certification working group of the North-East Asia Open-Source Software promotion forum. In their group activities, information on problems was shared by delegates from China, Japan, and Korea [4]. Furthermore, solutions to some problems were discussed in the collaborative work of the members [5].

A number of Web services can provide screenshots of many types of Web browsers. Moreover, a variety of studies, such as analyses of Web service interoperability [6] and browser-compatibility testing methods [7], have been conducted from the standpoint of software engineering.

Several studies on the standardization of Web content have been conducted over the past few years. Result of these studies too are useful for our work. Peter K. conducted massive studies of U.S. government Web sites [8] and the People's Republic of China government Web sites [9], using the W3C Validator. However, we consider his approach to be practically insufficient, because problems such as those discussed in this paper may not be detected by simply checking syntax errors.

6 Conclusions

We studied the interoperability-discrepancy problems of Web content for more than three years. Our activities resulted in the development of Pirkar, which helps Web designers easily create multi-browser Web applications.

Recently, browser vendors have become increasingly aware of standards. If all Internet users were able to access Web sites using the latest browsers, there would be no problem. However, it cannot be ignored that there are still many legacy users, who continue to use old-fashioned browsers that have interoperability problems, as mentioned in this paper. This is a strong motivation for developing the Pirka'r tool, which supports Web designers. All of the functions provided by Pirka'r are language-independent, so this tool can be used by Web developers worldwide. We will work in the future to promote this activity.

Acknowledgements

This study and Pirka'r development were conducted as part of the Open-Source Software-Utilization Development Program, supported by the Information Technology Promotion Agency, Japan.

References

1. B. Decrem, "Desktop Linux Technology & Market Overview," Open Source Application Foundation, July 2003.
2. IPA, "Research on the Improvement of Web Contents Compatibility Conducive to the Widespread Use of OSS Desktops, Resarch Report," `http://www.ipa.go.jp/software/open/ossc/download/Web_Research_En.pdf`, 2007.
3. IPA, "Research on the Improvement of Web Contents Compatibility Conducive to the Widespread Use of OSS Desktops, Written recommendations," `http://www.ipa.go.jp/software/open/ossc/download/Web_Recommendations_En.pdf`, 2007.
4. NEA-OSS Promotion Forum WG3, "Information Technology – Report of Web Interoperability," WG3 SWG2 N054, TR00003, 2007.
5. NEA-OSS Promotion Forum WG3, "Information Technology – Solution of Web Interoperability Discrepancy," WG3 SWG2 N063, TR00004, 2008.
6. V. De Antonellis and M. Melciori and P. Plebani, "An Approach to Web Service Compatibility in Cooperative Processes," *In Proceedings of SAINT 2003 Workshops,* pp.95–100, 2003.
7. L. Xu and B. Xu and C. Nie and H. Chen and H. Yang, "A Browser Compatibility Testing Method Based on Combinatorial Testing," *Lecture Notes in Computer Science,* Springer Berlin / Heidelberg, Vol.2722/2003, pp.310–313, 2004.
8. K. Peter, "Government Web standards usage: USA – standards-schmandards," `http://www.standards-schmandards.com/2005/government-web-standards-usage-usa/`, 2005.
9. K. Peter, "Government Web standards usage: People's Republic of China – standards-schmandards," `http://www.standards-schmandards.com/2006/gvmt-standards-prc/`, 2006.