PIRKA'R: TOOL FOR WEB DESIGNERS
Supporting Development of Multi-platform Web Application

Jun Iio, Hiroyuki Shimizu
Research Center for Information Technology, Mitsubishi Research Institute, Inc.
2-3-6 Otemachi, Chiyoda-ku, Tokyo, 100-8141 Japan

Hisayoshi Sasaki, Akihiro Matsumoto
Gluegent, Inc. 4-1-28 Toranomon, Minato-ku, Tokyo, 105-0001 Japan

Keywords: Web browser, Interoperability discrepancy, Commnity support, Web content designer.

Abstract: The elemental technologies of WWW have been standardized as HTML and CSS, recommended by W3C. Therefore, not only the servers but also web browsers are considered as components independent within the whole web application system so that they should be interchangeable. However, the actual web application has not yet led to this ideal situation. The problem of interoperability discrepancies between different implementation of web browsers still remains. So far we had studied the reason why this problem arose and collected instances of problem. Based on the result of our previous study, we developed an web-system designers' tool which can provide the useful information for them to avoid the pitfalls on the interoperability discrepancy problems.

1 INTRODUCTION

WWW has become a strong presence that represents modern information communicating infrastructure as a huge distributed knowledge environment. Various technologies were developed for the case where web browser was working as the client in cooperation with a wide variety of servers.

System components that consist of the WWW environment are categorized into functional layers, such as database-layer, application-layer, and presentation-layer. The specifications of protocol to connect each layer are strictly standardized so that it enables us to make modules for web application interchangeable.

In the ideal situation, these standards can increase the efficiency for system development. However, due to many practical reasons, almost all the web applications have not yet led to such an ideal condition. That is, each module has dependencies and system developers have to avoid conflicting and/or interoperability discrepancies between the modules required by them. As a result of that, there remains many severe problems, for example, expensive development due to vendor lock-in.

End-users of web applications are no exceptions as well. In the cases that modules are deeply depending on each other, the users have to keep using their legacy systems because such non-interchangeable modules prevent the users from migrating into a new system environment.

We have focused on the problems of the interoperability discrepancy between the different implementation of web browsers and studied to solve these problems. In this paper, we report the result of our study and present an overview of the tool that enables web developers to create multi-browser web application easily.

2 PROBLEMS TO SOLVE

Indeed web applications based on WWW technologies are supported by many standards, such as HTML, CSS, ECMAscript, and so on. Regarding any content that should be rendered in the web browser (except special items handled by plug-in component), its structure is described by HTML, XHTML, and DHTML, and its representation is defined by CSS. In addition, algorithm of dynamic content are written in

Iio J., Shimizu H., Sasaki H. and Matsumoto A.
PIRKAâĂŹR: TOOL FOR WEB DESIGNERS - Supporting Development of Multi-platform Web Application.
DOI: 10.5220/0002769701590162
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

JavaScript and DOM (Document Object Model). Recently it has became widely common to use AJAX technology for the user-interface layer.

W3C and ECMA recommended the specification for these technology, therefore it is expected that there is no difference among the behavior of any web browsers. However, there are many problems caused by the interoperability discrepancies between the different implementation of web browsers.

2.1 Problem Types

The problems are categorized into several types.

Typical problems are categorized as the use of an extra-function, which is implemented by browser vendors independently and it is not listed in the standard specification. Such a poor context is a sequel from "the browser war" that arose in more than ten years ago.

Other defects are caused from the implementation difference due to the ambiguousness in the specifications, bugs in software of the browser, and other various reasons.

Anyway, these problems force the end-user to use a specific browser. It is absolutely inconvenient for the users. Previously we reported our study that revealed the inconvenience of using non-interoperable web content in the education field.

2.2 Examples of Problems

A typical example of the unconformity problem is illustrated in Figure 1.

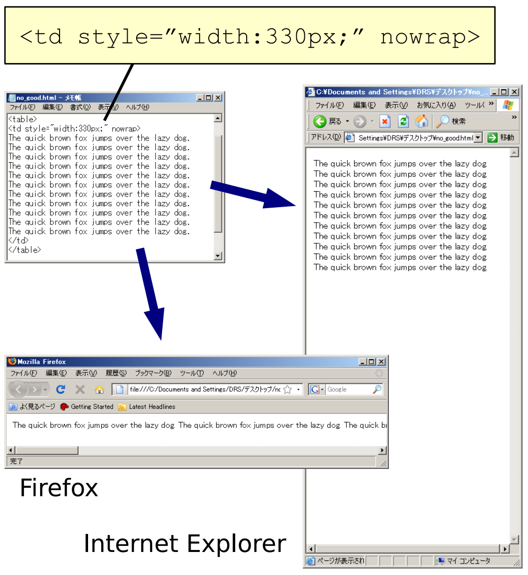


Figure 1: Confliction of "nowrap" and width definition.

In this example, the designer who wrote this HTML file intended to define column width with 330 pixels. In his/her plan, long sentences are supposed to be word-wrapped at the right end of the column. For some reason, `nowrap` attribute was unintentionally added. `Nowrap` attribute should be used to inhibit word-wrapping so that defining width and using `nowrap` attribute are in an inconsistent state.

According to the specification recommended by W3C, it is essentially correct to prioritize `nowrap` attribute over width definition. That is, in figure 1, the rendering result by Firefox is basically correct. However, Internet Explorer version 6 prioritizes width attribute over `nowrap`. Such a defect produces unwanted difference in rendering between each browser.

3 BACKGROUND OF DEVELOPMENT

These problems became serious, since the WWW had obtained an important position of the infrastructure in information society.

The Web Standards Project is working actively to ask for browser vendors to implement their software that is rigidly following the standard specifications. They prepare the Acid test in order to check whether the implementation of web browsers can meet the standards or not.

On the other hand, we adopted the other approach to solve this problem. We developed a system named TouchUpWeb[1], which can modify inappropriate content within the browser after downloading it from the Internet.

In our TouchUpWeb project, we tried to propose the appropriate environment for the end-users by solving problems within the user's web browser. The modification process is activated when the problems are found in our modification-database. To operate the system appropriately, we kept maintaining the database by collecting information about the web interoperability discrepancies and stored solution for that problems into the database.

Although TouchUpWeb was considered as an ad hoc approach, we were able to indicate some effectiveness of local modification under the certain situations. Furthermore we investigated how many problems existed in the Internet space and how they affected the web application users.

The result of our investigation (IPA, 2007a) revealed that about 170 interoperability problems potentially existed in the Internet. Some problems were

[1]This service was terminated at the end of 2008.

categorized into fatal error. For instance, some of them were never displayed in specific browser, and others were operated inappropriately in the browser. Based on this result, recommendation document (IPA, 2007b) was published. Table 1 shows the number of error factors classified by the impact of unconformity.

Table 1: Number of the detected interoperability discrepancies by impact.

Impact	# of factors	# of cases	% of detected cases in total
Large	14	3,309	3.0%
Medium	40	18,146	16.6 %
Small	51	69,169	63.4 %
None	13	1,178	1.1 %
N/A	48	17,366	15.9 %
Total	166	109,168	100.0 %

To improve this situation, we designed a tool supporting web designers, web developers, and web application programmers, to enable multi-browser web application developing easy and speedy.

The key point of this tool is information of the unconformities, offered to the developer to avoid the pitfall that disturbs the interoperability in their web applications. The system named "Pirka'r," which we report in this paper, was developed and was released as open-source software to the Internet. Now it is available for everyone under the Apache License, Version 2.0 from http://pirkar.ashikunep.org/.

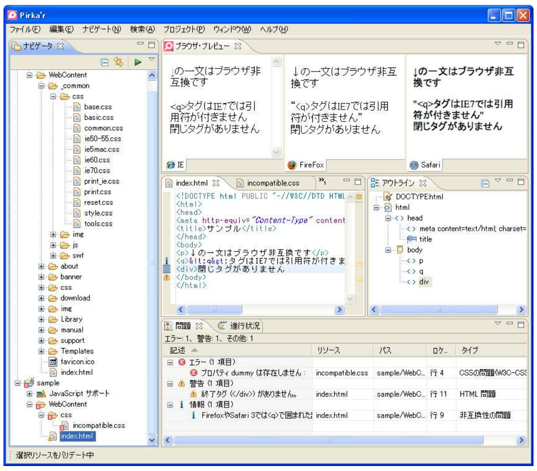
4 SYSTEM OVERVIEW

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

4.1 Overview of Pirka'r

Pirka'r consists in three parts, a client running on the user's workstation, a verification server, and a verification data management server. The client and verification server can be either installed in one workstation or in separated two computers.

Pirka'r has many functions that make help for web developers, such as verification against the standards, assessment of multi-browser interoperability, multi-browser previewing, automatic rendering, multifunctional editing, auto-downloading a set of web pages. Figure 2 shows a screenshot of Pirka'r running on Windows. A user of Pirka'r can oparete files shown in the left side pane, and can check multi-browser previews on the top of window. Multifunctional editor



↓の一文はブラウザ非互換です

<q>タグはIE7では引用符が付きません
閉じタグがありません

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML
<html>
<head>
<meta http-equiv="Content-Type" content
<title>サンプル</title>
</head>
<body>
<p>↓の一文はブラウザ非互換です</p>
<q>&lt;q&gt;タグはIE7では引用符が付きま
<div>閉じタグがありません
</body>
</html>
```

エラー: 1、警告: 1、その他: 1

記述	リソース	パス	ロケ...	タイプ
エラー (1 項目)				
プロパティ dummy は存在しません :	incompatible.css	sample/WebC...	行 4	CSSの問題(W3C-CSS
警告 (1 項目)				
終了タグ (</div>) がありません。	index.html	sample/WebC...	行 11	HTML 問題
情報 (1 項目)				
FirefoxやSafari 3では<q>で囲まれた	index.html	sample/WebC...	行 9	非互換性の問題

Figure 2: A screenshot of Pirka'r (Japanese version).

and outline processor are located in the middle row of the screen. The user can check error reports at the bottom row.

Pirka'r can help developers by verifing HTML / CSS / JavaScript description in his/her web content not to contain browser dependent descriptions.

The rendering result of target web pages are displayed in multiple browser panes. This function relieves the designers of annoying work handling a number of browsers and checking results of rendering by each browser [2].

4.2 Pirka'r Ecosystem

Figure 3 shows an overview of Pirka'r ecosystem.

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

We own the one management server that controls verification data and it can deliver the set of verification data to the verification engines that are installed in any local networks.

For the case that a designer find a new interoperability discrepancy problem, we have prepared the reporting form in our web site so that he or she has an opportunity to report the problem. After we received the report, committers in our team will try to create a new verification data that contains a check script and a series of document showing how to fix the problem.

[2]Currently multi-browser preview function is not available on Linux platform.

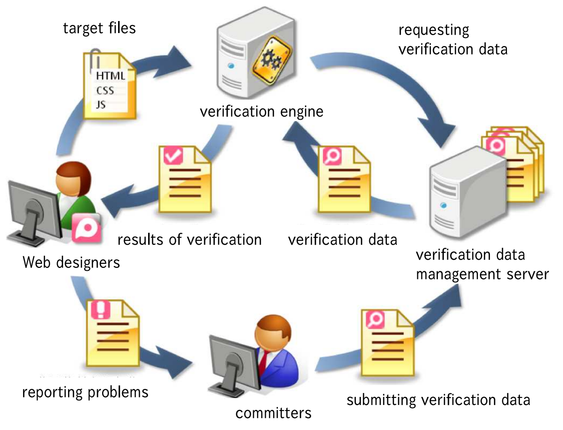


Figure 3: Overview of Pirka'r ecosystem.

4.3 Verification Script

The verification script is written in JavaScript. It searches defects in the target web content, traversing over its DOM.

The verification script is quite simple if the problem is not so complicated. We had already prepared more than 100 scripts to check the existing problems and they were stored in the master database.

Although we have more than 100 scripts currently, they are considered insufficient, since this problem is categorized as an error correction problem. As we described previously, there is a demand to add new verification data into the database from comitters.

5 RELATED WORKS

Opera web browser has the ”Opera's site patching” function, which enables its user to modify web content in Opera browser, based on quite similar idea to TouchUpWeb' method. Opera browser can modify downloaded content before rendering it, using Browser.js equipped in the browser.

A variety of studies such as analyses on the web service interoperability (Antonellis et al., 2003) or browser compatibility testing methods (Xu et al., 2004) have been conducted from the viewpoint of software engineering.

Several surveys on standardization of web content were conducted in a couple of years. The result of these surveys is also useful for our work. Peter K. conducted massive surveys against U.S. government websites (Peter, 2005) and People's Republic of China government websites (Peter, 2006). He used the W3C Validator for his survey. However, we consider that his approach is practically insufficient, because the real problems like what discussed in this paper cannot be detected only by checking syntax errors.

6 CONCLUSIONS

The interoperability discrepancy problems regarding web content were shown in the former section of this paper. And the system Pirka'r to solve these problems was proposed. Pirka'r can help web designers in designing multi-browser web application easily.

Modern browsers tend to have better interoperability than before, however it is not ignorable that there are many legacy users who keep using the old-fashioned browsers, which have some problems in interoperability that we mentioned in this paper. That is the reason why we need to develop this tool, Pirka'r.

ACKNOWLEDGEMENTS

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

REFERENCES

Antonellis, V. D., Melciori, M., and Plebani, P. (2003). An approach to web service compatibility in cooperative processes. In *Proceedings of SAINT 2003 Workshops*, pages 95–100.

IPA (2007a). 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.

IPA (2007b). 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.

Peter, K. (2005). Government web standards usage: Usa – standards-schmandards. http://www.standards-schmandards.com/2005/government-web-standards-usage-usa/.

Peter, K. (2006). Government web standards usage: People's republic of china – standards-schmandards. http://www.standards-schmandards.com/2006/gvmt-standards-prc/.

Xu, L., Xu, B., Nie, C., Chen, H., and Yang, H. (2004). A browser compatibility testing method based on combinatorial testing. In *Lecture Notes in Computer Science*, volume 2722/2003, pages 310–313. Springer Berlin / Heidelberg.