International Journal of Scientific Research in Computer Science and Engineering
Volume-5, Issue-1, pp.1-6, February (2017)

Research Paper

E-ISSN: 2320-7639

Detection of Cross Browser Inconsistency by Comparing Extracted Attributes

C.P.Patidar[1], Meena Sharma[2], Varsha Sharda[3*]

[1]Information Technology, IET, DAVV, Indore, India
[2]Computer Science, IET, DAVV, Indore, India
[3]Computer Science, Medicaps, Indore, India

Available online at: www.isroset.org

Received 28th Dec 2016, Revised 12th Jan 2017, Accepted 02nd Feb 2017, Online 28th Feb 2017

***Abstract*—The advancement in web technology and popularity of web applications amplifies the inconsistencies between various web browsers. These inconsistencies augment cross browser incompatibilities that constitute different look on different browsers for a particular web application. In some cases, Cross-Browser Inconsistencies (XBIs) consists of acceptable difference whereas these may entirely prevent users from accessing part of a web application's functionality in other cases. Therefore, testing process of a web application must be performed comprehensively on multiple browsers to achieve consistency. Available tools and techniques require a considerable manual effort to recognize such issues and provide limited support for fixing the cause of the issues. In this paper, we propose a technique for detecting cross-browser issues without human intervention.**

***Index Terms*: Browser, Cross Browser Inconsistency, Reliability, Web application**

I. INTRODUCTION

Presently, web applications are evolved from web systems or websites based on a client server model. When a client issues a request to the server through a web browser then the server side components get invoked. These communications generate requests to the server, and the server responds to such requests with updates to the current web page, programmed in HTML (Hyper-Text Mark up language) or XML (extensible Mark-up Language), and to other related resources, such as style information in CSS (Cascading Style Sheets), client-side code (e.g., JavaScript), images, and so on. Subsequently, these resources are used to calculate and render an updated web page in the web browser. Web applications often have variable elements such as advertisements and generated content (e.g., time, date etc.) which are dissimilar across multiple requests. If these elements are not ignored, the technique might consider these as changes across browsers thus resulting in false positives in the results. Hence, the technique requires discovering and leaving out such elements during comparison.

A web browser is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. By a Uniform Resource Identifier (URL), a web page, image and video an information resource is recognized. The browser gets in contact with the web server and needs for information. The web server receives the information and displays it on the computer. The major problems associates with using the web application through different web browsers are related with web browser inconsistency. Also, web applications are being used by many for all activities in every field of work. Some variation in arrangement of elements or content of a web-based application on different browsers is known as Cross-Browser Inconsistency .When a user execute a web application on multiple browsers, then some web application exhibit different behaviours and thus introduces Cross-Browser Inconsistencies (XBIs) [1]. XBIs exhibit differences between a web application's appearances, behavior, or both, when it is executed on two different environments. If cross browser inconsistencies are not being correctly tested during the testing phase, then it can negatively affect the experience of the user of web application. Consequently, identification of the cross browser inconsistencies is an essential factor and is a serious concern for companies dependent on such applications for business. Rapidly changing technologies have accordingly driven up the number of web browser version release and browsers are the main interfaces to deliver/access the information in one click [2].

It seems that, website is developed using one browser rather than for multiple browsers. Testing across a variety of browsers will expose issues the developer may be unaware of. Accordingly, we performed a systematic study

*Corresponding Author: Varsha Sharda
E-mail: varsha.2822@gmail.com, Tel.: +91 7354531265

on various real-world web applications. This study facilitates us to found a categorization of XBIs that aids in defining our technique. We found three major varieties of XBIs: structural, content and behaviour.

(i) ***Structural XBIs****:* This kind of XBI affects the structure, or arrangement, of individual web pages. The web page structure is basically a particular arrangement of elements, which in case of structural XBIs is incorrect in a particular browser. For instance, the misalignment of one or more elements on a specified web page, in a particular browser, can comprise a structural XBI [3].

(ii) ***Content XBIs:*** This type of XBI is examined in the content of individual components on a web page. Such differences can take place, where the graphical appearance of a web page element, or the textual value of an element, is different across two browsers. We further categorize this type of inconsistency as visual-content and text-content XBIs [3].

(iii) ***Behavioural XBIs:*** These types of XBIs involve differences in the behaviour of individual widgets on a web page. An example of such an XBI would be a button that performs a particular action within one browser and a totally different action, or no action at all, in another browser [3].

In addition, the internet of web application has quietly become one of the important medium of the business. The software faults in web applications have potentially leads to the failure or the underperformance of the business. Most of the works on web applications have been on making them more powerful. But, quite little is done to guarantee the quality. Key quality attributes for web applications include reliability, availability, interoperability and security apart from ensuring the functional & usability aspects [4].

Web browser compatibility testing is technical and puzzling - something you have to let your web developer deal with. The problem is that if your website is not well-suited with the plethora of browsers available, it will impinge on your business reputation [5].

The recent work on identifying XBIs has proposed techniques that focus only on certain aspects of a web application's execution, and are well appropriate for specific types of XBIs. For example, the WebDiff tool uses computer vision to detect XBIs, whereas CrossT uses graph isomorphism along with text comparison to find XBIs [6]. These tools provide only partial and imprecise solutions to the XBI detection problem.

To address the drawbacks of existing techniques, we proposed a technique that integrates a rich set of comparison techniques and orchestrates them to apply each technique to the category of XBIs that it is best matched to detect. Our technique is a computerized, defined, and widespread approach for XBI detection.

The key contributions of this work are:

- A new technique and tool for detecting both visual and structural XBIs in web applications.
- An innovative, powerful technique to detect visual XBIs.
- An evaluation of this technique on several real-world web applications that shows its effectiveness in detecting different kinds of inconsistencies XBIs.

The rest of the paper is organized as follows. Section I contain Introduction of cross browser inconsistency along with web application ,Section II contain related work done in area of cross browser inconsistency of a web application, Section III describes problem definition of our research work .Section IV explains our proposed solution to detect cross browser inconsistency with flow chart, Section V contain application area of our research work, Section VI contain expected outcomes of our proposed methodology and Section VII concludes research work with future directions.

II. RELATED STUDY

With the proliferation of several browser versions and release updates, testing infrastructure requirements are no longer static. Given the various smart devices flooding the market each day, cross-browser compatibility has emerged as a major challenge for software testers.

It has been observed that, TCS puts a light on accomplishment of Cross Browser Testing Tool. It offers a computerized solution with a preconfigured collection of devices and test environment that enables quick testing across multiple OS and browsers. It connects to real devices for testing of web based mobile applications and ensures precise cross browser and cross device testing. It covers three major areas of testing: Cross browser UI validation, Functional testing to ensure the accuracy of functionality, RWD testing to address inconsistencies in pages while ensuring best possible viewing and user interaction across a wide range of devices. They also offers a solution using efficient layout comparison, functional test, responsive web design, broken link validation and portal based management for cross browser supports and gains up to 60% savings in test effort through dynamic script creation. It gives an enhanced UI scanning method and centralized test management. Almost, 50% time saving with automatic PDF evaluation and parallel test implementation across multiple browser versions [7].

In addition, the details of widespread approach for cross compatibility testing of website have been specified. It covers the technical complexities of a website and differences in the browsers, operating systems, and devices require detecting cross browser inconsistency. Also, the parameters that a website must meet before its launching worldwide and some cross browsers automated testing tools which assist in website testing on a variety of browsers, operating systems and devices and meet technical requirements for assuring website quality have been

suggested. Subsequently, a range of tools on the basis of their speed, pricing mode, interfaces, delays, scroll bars and additional features have been compared. Accordingly, the web performance testing for web site functionality on different web browsers, operating systems and different hardware platforms is checked for software, hardware memory leakage errors [8].

Next, the relative study of cross browser compatibility as design issue in different web sites based on an online tool using .NET Framework has been devised. It provides different development and design issues in various kinds of websites like Government Websites, Educational Websites, Commercial Websites, Social Networking Websites, and Job Portal Websites. The results obtained after testing five different categories of websites shows that educational and social networking sites displays slightest compatibility in multiple browsers where as job portals, commercial and government websites shows 100% observance to the website design principles suggested by W3C [9].

Moreover, to make the browser a protected environment for running programs by introducing a separation method that insulates one application from the performance of another has been presented [10]. It shows the use OS processes within the browser to safely separate programs in a mode that is both efficient and backwards compatible with existing web sites. Also, it recognize content on the web page whether it is more active content or rich active content along with the trouble with browsers like failure separation, concurrency and memory management. Through measurements of both web content and browser behavior, these have shown that current web browsers provide unpredictable surroundings for running applications. This leads to serious problems with respect to failure isolation, concurrency, and memory management. These have shown that browser-based applications can be safely isolated from each other using OS processes. Processes prevent unwanted communications between programs in the browser, and they are well-organized relative to other browser operations, both in time and memory overhead.

Then, quantitative categorization of browser vulnerabilities to project the numbers of vulnerabilities for mapping test and improvement resources more efficiently have been presented [11]. Vulnerability discovery data for the three key browsers, Internet Explorer, Firefox and Mozilla have been examined and fixed to a vulnerability discovery model, and the integrity of fit is statistically examined. It also classifies Vulnerabilities based on cause, severity, impact and source. Classification of Vulnerability such as Input Validation Error (includes boundary condition error, buffer flood), Access Validation Error ,Exceptional situation Error ,Environmental Error, Configuration Error, Race Condition Error, Design Error.

Later, the difficulty of cross-browser compatibility testing of web applications as a functional consistency check of web application behavior across different web browsers with an automated solution have been posed [12]. This approach consists of automatically analyzing the given web application under different browser environments and capturing the behavior as a finite-state machine and comparing the generated models for equivalence on a pair wise basis and exposing any observed discrepancies. This overall approach consists of a two-step process. The first step is to automatically crawl the given web application under multiple browser environments and capture and store the observed behavior, under each browser, as a specific state machine navigation model. The crawling is done in an identical fashion under each browser to replicate precisely the same set of user interaction sequences with the web application, under each environment. The second step consists of formally comparing the generated models for similarity on a pair wise-basis and revealing any experimental discrepancies.

Further, a tool for identifying XBIs in web applications automatically, without requiring any effort from the developer has been provided [13]. This tool can work with any web application that runs on desktop browsers. This model captures screen and then compares the graph generated by crawler by graph isomorphism checking method. Also, it identifies different types of inconsistencies in a web application. It also generates easy to understand and actionable reports for the developer, thus allowing them to deal with XBIs more efficiently.

Accordingly, a thread level study of the work load generated by Google's Chrome browser on a heterogeneous multi-processing (HMP) platform found in numerous smart phones have been presented [14]. The thorough traces of the thread workload generated by the web browser, especially the rendering engine examined by it, and discuss the power saving potentials in relation to power management policies in Android. It also emphasizes on power management of web browser workload characterization on HMP platforms and seeks potential power savings based on the interpretation. Focus on the actual thread workloads and a function call invoked by the web browser is the new feature offered by it. It also provides the information that can be used for power management. Then, various tools enabling parallel execution of a range of automated tests using several remote test environments with different web browsers have been recommended. It also presents a tool for automated testing of web applications based on the Selenium RC framework.

III. PROBLEM DEFINITION

A high-quality web design aims to offer an identical appearance to the website viewed from any web browser. Consequently, a good quality website must be viewable in its

complete functionality on any web browser. As every webpage is made up of a range of components with its own uniqueness and it affects the performance of a webpage in different contexts. Similar to other parameters of performance assessment the browser compatibility aspect of website is also affected by different components of a webpage either directly or indirectly. In addition, different technologies produce the compatibility issue. As a result, for the period of the design stage of the websites these must be tested meticulously for its compatibility at different browsing environments. Section I discussed about the parameters that can affect the cross browser inconsistency of a web application.

IV. PROPOSED SOLUTION

Description

To identify cross browser inconsistencies, we propose a model to detect XBI. Figure 1 depicts an overview of our proposed XBI detection technique that takes as input the URL of the home page of the web application under test, URL and two browsers considered for the testing, Browser1 and Browser2. It produces output as list of identified inconsistencies. Our proposed model compares extracted attributes from crawler generated graph.

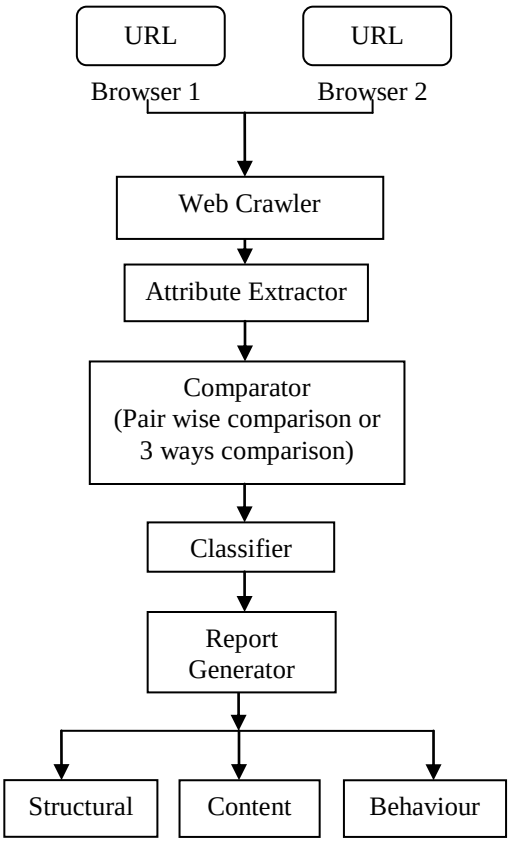

FIGURE 1: MODEL FOR CROSS BROWSER INCONSISTENCY

(i) Web Crawler

A web crawler is an automated program, or script, which methodically scans or "crawls" through web pages to generate an index of the data it is set to look for. This process is called as Web crawling or Spidering. We proposed to use a web crawler called "WebSPHINX (Website-Specific Processors for HTML Information extraction)" written in java. It crawl different websites and produces graph for that. It is open source web crawler and source code is available at websphinx.zip.

(ii) Attribute Extractor

Attribute extractor is based on the following hypothesis: In a graph generated by web crawler, since attribute terms repeat in multiple graphs for a web application, they are more likely to occur than other terms. We try to exploit this redundancy to capture the attributes. Thus the simplest way to select attributes would be to take the most frequent terms in the graphs. However, this method has a drawback. This method gives only frequent attributes and is likely to neglect rare attributes appearing in only few graphs. To overcome the first problem, we propose a two stage method. In the first stage, we cluster the all the words found in the graphs such that all the words close to an attribute are grouped together in a single cluster. This results in word clusters of different sizes. In the second stage, we extract an attribute from each cluster.

(iii) Comparator

This module performs textual analysis of corresponding elements to detect text-content XBIs. For detecting image-content XBIs, it compares screen images of the corresponding elements on the web page. The structure of the page extracted by the crawler is analyzed by the Layout Analysis component to create alignment graphs, which represent the relative alignment of web page elements. Comparison can be either pair wise where two attributes from graph are compared or it can be 3 way comparisons where three attributes extracted from graph are compared.

(iv) Classifier

This module classifies the type of inconsistency present in a web application according to its type that whether it is structural, content or behavioral inconsistency.

(v) Report Generator

This module generates a report written in HTML tabulates the set of detected XBIs.

V. APPLICATION

Application area of detecting cross browser inconsistency is the E-commerce websites, commercial websites, educational websites, government websites, news portal, and social networking websites. For the same, there exists a requisite

that a web application must behave similarly when executed on multiple different browsers.

VI. EXPECTED RESULTS

It has been observed that, when a web application is executed on multiple browser then expected outcome of our proposed model is to identify three main types of inconsistencies, if exists. This proposed model also generates report of inconsistencies. As this technique uses three-way comparator that compares three graphs generated by crawler simultaneously. Therefore, finding of XBIs can be fast technique as compared to other available tools. These are illustrated as below:

- We found that, structural XBIs are the most general class of XBIs, happening in 57% of the subjects with XBIs.
- We ascertain that these content XBIs occurred in 30% and 22% of the sites with XBIs respectively.
- We determine that behavioural XBIs occurred in 9% of the web applications with XBIs.

Thus, we can conclude that the behavioural XBIs have an effect on the functionality of individual components, resulting in broken navigation between different screens. On the other hand, Structural and content XBIs involve differences in the arrangement or depiction of elements on a particular web page.

VII. CONCLUSION

XBIs are a severe problem for web developers. Existing research tools only target particular aspect of XBIs and can report a significant number of false positives and negatives. To deal with these limitations, we presented our proposed model for detection of XBI.To accomplishes this task this paper presents the overview of proposed technique, its application and expected results. In addition, it also generates easy to understand reports for the developer, therefore allowing them to deal with XBIs more effectively. This is a challenging difficulty, as the application will necessarily look different in the two platforms, but it should offer the same, or at least similar functionality.

REFERENCES

[1] C.P.Patidar and Meena Sharma ,"An automated approach for cross browser inconsistency(XBI) detection", Ninth annual ACM India conference organized by ACM India, Oct 21-23,2016.

[2] Nepal Barskar, C.P.Patidar and Meena Sharma, "Analysis and Identification of Cross Browser Inconsistency Issues in Web Application using Automation Testing", International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555Vol.6, No3, May-June 2016.

[3] Nepal Barskar and C.P. Patidar, "A Survey on Cross Browser Inconsistencies in Web Application", International Journal of Computer Applications (0975 – 8887) Volume 137 – No.4, March 2016.

[4] "WebTesting", mindlance.com, testing@mindlance.com.

[5] Ochin and Jugnu Gaur, "Cross Browser Incompatibility: Reasons and Solutions", International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.3, July 2011.

[6] Shauvik Roy Choudhary, Husain Versee and Alessandro Orso, "WEBDIFF: Automated Identification of Cross-browser Issues in Web Applications", 26th IEEE International Conference on Software Maintenance in Timisoara Romania, 978-1-4244-8628-1/10, 2010.

[7] http://www.tcs.com/assurance, 2016.

[8] Sanjay Dahiya1, Ved Parkash1 and T.R. Mudgal2, "Comprehensive Approach for Cross Compatibility Testing of Website ", National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing (RTMC) ,Proceedings published in International Journal of Computer Applications (IJCA) ,2011 .

[9] Jatinder Manhas," Comparative Study of Cross Browser Compatibility as Design Issue in Various Websites" , BIJIT - BVICAM's International Journal of Information Technology Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA), NOV 2014.

[10] Charles Reis, Brian Bershad, Steven D. Gribble and Henry M. Levy, "Using Processes to Improve the Reliability of Browser-based Applications", University of Washington Technical Report UW-CSE, DEC 2007.

[11] Sung-Whan Woo, Omar H. Alhazmi and Yashwant K. Malaiya,"AN ANALYSIS OF THE VULNERABILITY DISCOVERY PROCESS IN WEB BROWSERS", proceeding of the 10th IASTED International Conference Software engineering and Applications, Dallas,TX,USA,ISBN: 0-88986-642-2 / CD: 0-88986-599-X, NOVEMBER 13-15, 2006,

[12] Ali Mesbah and Mukul R. Prasad," Automated Cross-Browser Compatibility Testing", ICSE '11,Waikiki, Honolulu, HI, USA ,ACM 978-1-4503-0445-0/11/05, May 21–28, 2011 .

[13] Shauvik Roy Choudhary, Mukul R. Prasad and Alessandro Orso," X-PERT: A Web Application Testing Tool for Cross-Browser Inconsistency Detection", ISSTA'14,San Jose, CA, USA,Copyright 2014 ACM 978-1-4503-2645-2/14/07,July 21–25, 2014.

[14] Nadja Peters1, Sangyoung Park1, Samarjit Chakraborty1, Benedikt Meurer2, Hannes Payer2 and Daniel Clifford2, "Web Browser Workload Characterization for Power Management on HMP Platforms", CODES/ISSS '16 Pittsburgh, PA, USA, ACM, and ISBN: 978-1-4503-4483-8/16/10, October 01-07 2016.

[15] Shauvik Roy Choudhary, Mukul R. Prasad and Alessandro Orso, "CROSSCHECK: Combining Crawling and Differencing to Better Detect Cross-browser Incompatibilities in Web Applications", 2012.

[16] Shauvik Roy Choudhary,"Detecting Cross-browser Issues in Web Applications", ICSE '11, Waikiki, Honolulu, HI, USA, ACM 978-1-4503-0445-0/11/05, May 21–28, 2011.

Authors Profile

C.P.Patidar received the B.E. degree in information technology and M.E. degree in computer engineering. He is an assistant professor of Information Technology at the Devi Ahilya University, Indore, India. His research interests are in cross browser inconsistencies, GPGPU computing, CUDA programming multithreaded architecture and memory architecture of computers.

Meena Sharma received the B.E. degree in computer engineering and M. Tech degree in computer science in 1992 and 2004 respectively. She received the Ph. D. Degree in computer engineering in 2012.She is a professor of Computer Engineering at the Devi Ahilya University Indore, India. Her research interests are in software engineering, software quality matrices and object oriented modelling and design.

Varsha Sharda received the B.E. degree in computer engineering.She is an assistant professor of Computer Science and engineering at the Medi-caps University Indore, India.Her research interests are in software engineering, Database management and object oriented analysis and design.

