Position Paper: Improving Browsing Environment Compliance Evaluations for Websites

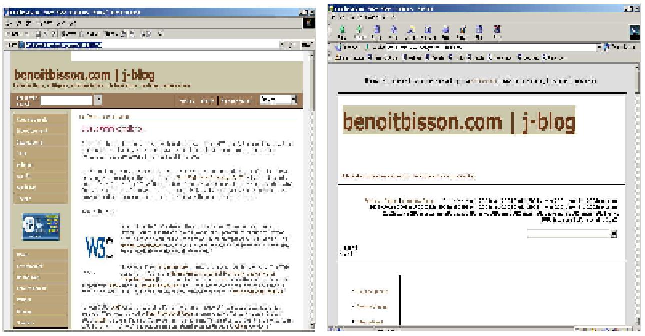
Cyntrica Eaton and Atif Memon
Department of Computer Science
University of Maryland, College Park, MD 20742
{ceaton, atif} @cs.umd.edu

ABSTRACT

Though it would be ideal for web pages to render and function consistently across heterogeneous browsing environments, the browser, browser version, and operating system used to navigate and interact with web content is known to have a significant impact on the subsequent level of user accessibility. While research endeavors directed toward improving web accessibility have generally focused on addressing usability issues for individuals with physical limitations, providing accessible information and services for the entire web population also encompasses addressing the limitations of devices and platforms used to deploy web pages. We propose that more research be invested in the latter issue to facilitate the development of effective tools for detecting browsing environment influenced usability issues before inaccessible pages are released in the field.

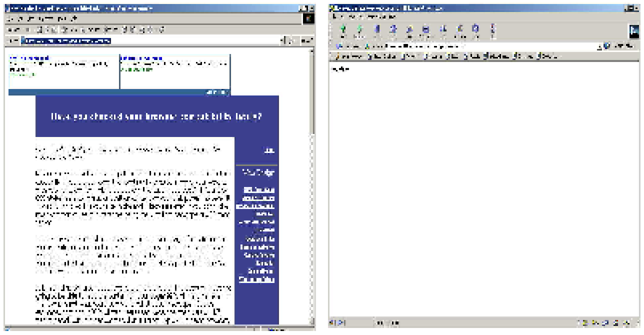
1. PROLOUGE

To further support the need for tools that will detect existing browsing environment accessibility barriers and alert web developers accordingly, consider the following examples of web pages rendered in two different environments. In each of the screenshots that follow, the results of launching the featured page in Internet Explorer are shown on the left and the effects of deploying the page in Netscape are shown on the right.



(a) (b)

Figure 1: [1] shown in Internet Explorer 6.0 (a) and Netscape 4.7 (b)



(a) (b)

Figure 2: [2] shown in Internet Explorer 6.0 (a) and Netscape 4.7 (b)

These are essentially two of many examples where users with different browser, browser version, and operating system combinations can experience a dramatic imbalance in web page appearance and performance. We encourage you to launch the pages featured in Figures 1 and 2 ([1], [2]) in various browsing environments to experience the differences in content presentation firsthand.

2. INTRODUCTION

Increased reliance on the benefits that stem from a globally interconnected system coupled with the demands and expectations of the growing web community have collectively driven a relevant research effort directed toward improving all aspects of web technology. Addressing the issues that threaten to diminish universal accessibility for the diverse web audience, in particular, is one facet of this overall endeavor. While there is a significant amount of work invested in improving accessibility for web constituency with sensory, cognitive, and physical limitations, another important, yet less heavily studied, web usability factor is the accessibility constraints imposed by end-user browsing environments[3]. More specifically, the use of diverse, heterogeneous combinations of browser, browser version, and operating system can have a significant impact on web page presentation and functionality among users. Establishing evaluation techniques that can effectively identify browsing environment specific accessibility issues will simultaneously empower web developers to avoid hindering users from exploring and interacting with the information and services featured on a website unwittingly and increase the potential of productive sessions for web users.

3. APPROACHES

Intuitively, one approach to evaluating the usability and performance of a web site in target environments is to perform dynamic testing by manually launching pages and observing subsequent presentation and functionality. While this testing strategy allows evaluators to experience, firsthand, any accessibility issues that exist, limitations on time, manpower, and computing resources can severely restrict the depth of the website tested and the breadth of browsing environments explored. Consequently, most execution-based evaluations limit testing environments to several versions of Internet Explorer and Netscape on Windows and Mac OS. This, of course, excludes many other prospective client platform configurations from the testing process and, subsequently, does not provide an adequate foundation for establishing confidence in universal usability.

An alternative, highly effective quality assurance evaluation that can be used to assess web page interoperability across varied client environments is the code review. Briefly, the basis of a code review is static analysis aimed at identifying fragments of code consistently associated with faulty behavior. In the web realm, the source of these faulty code fragments, or bug patterns, is relatively straightforward; browsing environment related obstacles arise when unrecognized HTML tags are encountered in document source code. More specifically, a page that renders correctly in one browsing environment may be significantly defective in another based on the relative support of the tags contained in the document source. Consequently, HTML tags are important accessibility predictors when support for a given tag is known to be nonexistent or insufficient; evaluating the compliancy of a web page within an environment can be reduced to identifying browsing environment specific bug patterns. The strength of such an evaluation, however, would be heavily reliant on the comprehensiveness, or completeness of the tag compliancy rule set. To further illustrate these ideas, consider the following definitions:

Definition 1: Bug Patterns

Let **E** denote a browsing environment defined by the triplet <**B**, **V**,**O**> where **B** is the browser, **V** is the browser version, and **O** is the operating system. Consider **T**, the universal tag space of all possible HTML document source tags:

$$\{\forall\, e_j \in \mathbf{E}\ \exists\ \mathbf{I} = \{i_1, i_2, \ldots i_{|\mathbf{I}|}\} \text{ s.t. } (\mathbf{I} \subseteq \mathbf{T}) \wedge (\text{unsupported } (\mathbf{I}, e_j))\}$$

That is, each browsing environment supports only a portion of the overall tag space. All others are unrecognized, or incompliant, in the targeted browsing environment. Consequently, the tags in **I** could be considered to be bug patterns for web pages rendered in environment e_j and cross-browser accessibility of a web site can be evaluated in a code review by detecting the presence of tags $i_n \in \mathbf{I}$.

Example 1:

The tag <div>, though a part of the comprehensive tag set **T** and supported by Internet Explorer, is unrecognizable in browsing environments that feature Netscape. Consequently, <div> would be an element of **I** for environments **E** where **B** in the corresponding triplet is Netscape.

Definition 2: Bug Pattern Completeness

The accuracy of code reviews based on the set of tags in **I** is largely dependent upon the accuracy and completeness of the description for **I**. If for instance:

$$\{\exists\ t_i \in \mathbf{T} \text{ s.t. } (t_i \notin \mathbf{I}) \wedge (\text{unsupported}(t_i, e_j))\}$$

performance of a static analyzer that does not include t_j as a bug pattern is compromised.

Example 2:

Consider <blink>, a tag that is unsupported in browsing environments featuring Internet Explorer. If compliance evaluation was executed for a web page that incorporated the <blink> tag yet the tag was not listed in **I** as a bug pattern, the accuracy of the resulting report would be compromised, and developers would be subject to latent failures and a false confidence in compliance.

Given the usefulness of bug patterns in evaluating environment compliance and the need for comprehensive pattern sets, it is our position that developing tools which identify the presence of environment specific bug patterns in web pages and support mechanisms for updating and refining knowledge of tag compliance rules is a first step in improving web quality for users when browsing environment influenced accessibility issues are an issue.

4. RESEARCH DIRECTIONS

In an effort to address the accessibility challenges influenced collectively by the browser, browser version, and operating system used for web navigation, we have developed a tool that estimates web page interoperability based on knowledge of the HTML tags that comprise web pages and knowledge of browsing environment specific tag support. As a result, one research direction inherent in our approach is discovering efficient ways of comparing web page source code to a list of bug patterns or, in this case, unsupported HTML tags. These lists can be relatively long for each browsing environment. Determining efficient ways of representing the rules for each environment and conducting subsequent comparisons efficiently is fundamental to the success of any tool featuring this approach.

As noted before, the strength of a code review-based evaluation strategy is heavily dependent upon the comprehensive nature of tag support knowledge for various browsing environments. In other words, conducting code reviews with truncated or inaccurate knowledge can severely inhibit accurate compliance analysis, causing a page to appear as if it will render properly in an incompliant environment. Consequently, another research direction would be to identify effective ways of updating tag support knowledge as a means of ensuring that all incompliant tags are represented in the bug pattern set associated with a given browsing environment. One method of optimizing knowledge of tag support criteria is to design a learning mechanism that can estimate the compliance of a novel HTML tag, or a tag for which compliance or incompliance knowledge has not been established, based on observations of positive (accessible) and negative (inaccessible) examples of page presentation and functionality. The underlying theory of such a technique is that observation of HTML tags that are positively correlated with inaccessible web pages can provide insight into the root causes of inaccessibility.

The overall endeavor to detect browsing environment accessibility barriers and to ensure comprehensive knowledge of tag support fall under general research endeavors associated with bug isolation [4] and bug patterns[5]. In this case, bug isolation is a related endeavor because the attempt to discover new tag support rules is largely an endeavor to isolate faulty tags given observation of source documents. Bug patterns, of course, are related because faulty tags can be recast as bug patterns and used during a code review to detect possible accessibility barriers. Research directions in which measures derived from fundamental principles of either of these two concepts are also viable.

5. CONCLUSION

Enabling consistent rendering of web pages across heterogeneous browsing environments is one facet of a general endeavor to support and improve the quality of user experiences on the web. From our prospective, providing tools capable of detecting browsing environment accessibility issues is extremely important for supporting general web quality and there are numerous research directions entailed in developing effective solutions.

6. REFERENCES

[1] http://benoitbisson.com/blog/archives/000180.html#000180

[2] http://www33.brinkster.com/articlesupport/browser.html

[3] Hull L (2004) Accessibility: It’s not just for disabilities any more, Interactions **11**(2): 36-4

[4] Liblit B, Aiken A, Zheng A, Jordan M (2003) Bug Isolation via Remote Program Sampling, Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation: 141-154

[5] Allen E., Bug Patterns: An Introduction: Available at: htttp:///www-106.ibm.com/developersworks/library/j-diag1.html(2001)

Position Paper: Improving Browsing Environment Compliance Evaluations for Websites

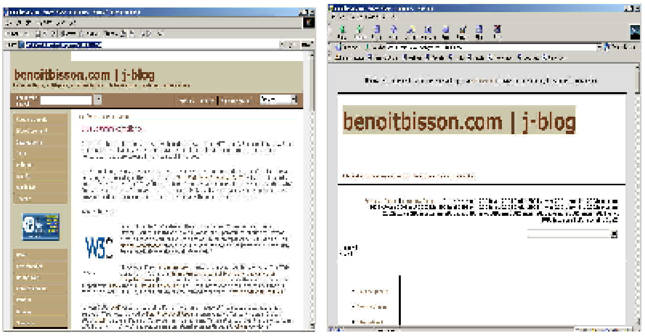
Cyntrica Eaton and Atif Memon
Department of Computer Science
University of Maryland, College Park, MD 20742
{ceaton, atif} @cs.umd.edu

ABSTRACT

Though it would be ideal for web pages to render and function consistently across heterogeneous browsing environments, the browser, browser version, and operating system used to navigate and interact with web content is known to have a significant impact on the subsequent level of user accessibility. While research endeavors directed toward improving web accessibility have generally focused on addressing usability issues for individuals with physical limitations, providing accessible information and services for the entire web population also encompasses addressing the limitations of devices and platforms used to deploy web pages. We propose that more research be invested in the latter issue to facilitate the development of effective tools for detecting browsing environment influenced usability issues before inaccessible pages are released in the field.

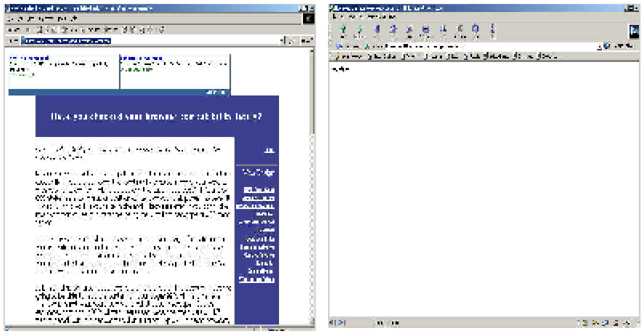
1. PROLOUGE

To further support the need for tools that will detect existing browsing environment accessibility barriers and alert web developers accordingly, consider the following examples of web pages rendered in two different environments. In each of the screenshots that follow, the results of launching the featured page in Internet Explorer are shown on the left and the effects of deploying the page in Netscape are shown on the right.



(a) (b)

Figure 1: [1] shown in Internet Explorer 6.0 (a) and Netscape 4.7 (b)



(a) (b)

Figure 2: [2] shown in Internet Explorer 6.0 (a) and Netscape 4.7 (b)

These are essentially two of many examples where users with different browser, browser version, and operating system combinations can experience a dramatic imbalance in web page appearance and performance. We encourage you to launch the pages featured in Figures 1 and 2 ([1], [2]) in various browsing environments to experience the differences in content presentation firsthand.

2. INTRODUCTION

Increased reliance on the benefits that stem from a globally interconnected system coupled with the demands and expectations of the growing web community have collectively driven a relevant research effort directed toward improving all aspects of web technology. Addressing the issues that threaten to diminish universal accessibility for the diverse web audience, in particular, is one facet of this overall endeavor. While there is a significant amount of work invested in improving accessibility for web constituency with sensory, cognitive, and physical limitations, another important, yet less heavily studied, web usability factor is the accessibility constraints imposed by end-user browsing environments[3]. More specifically, the use of diverse, heterogeneous combinations of browser, browser version, and operating system can have a significant impact on web page presentation and functionality among users. Establishing evaluation techniques that can effectively identify browsing environment specific accessibility issues will simultaneously empower web developers to avoid hindering users from exploring and interacting with the information and services featured on a website unwittingly and increase the potential of productive sessions for web users.

3. APPROACHES

Intuitively, one approach to evaluating the usability and performance of a web site in target environments is to perform dynamic testing by manually launching pages and observing subsequent presentation and functionality. While this testing strategy allows evaluators to experience, firsthand, any accessibility issues that exist, limitations on time, manpower, and computing resources can severely restrict the depth of the website tested and the breadth of browsing environments explored. Consequently, most execution-based evaluations limit testing environments to several versions of Internet Explorer and Netscape on Windows and Mac OS. This, of course, excludes many other prospective client platform configurations from the testing process and, subsequently, does not provide an adequate foundation for establishing confidence in universal usability.

An alternative, highly effective quality assurance evaluation that can be used to assess web page interoperability across varied client environments is the code review. Briefly, the basis of a code review is static analysis aimed at identifying fragments of code consistently associated with faulty behavior. In the web realm, the source of these faulty code fragments, or bug patterns, is relatively straightforward; browsing environment related obstacles arise when unrecognized HTML tags are encountered in document source code. More specifically, a page that renders correctly in one browsing environment may be significantly defective in another based on the relative support of the tags contained in the document source. Consequently, HTML tags are important accessibility predictors when support for a given tag is known to be nonexistent or insufficient; evaluating the compliancy of a web page within an environment can be reduced to identifying browsing environment specific bug patterns. The strength of such an evaluation, however, would be heavily reliant on the comprehensiveness, or completeness of the tag compliancy rule set. To further illustrate these ideas, consider the following definitions:

Definition 1: Bug Patterns

Let **E** denote a browsing environment defined by the triplet <**B**, **V**,**O**> where **B** is the browser, **V** is the browser version, and **O** is the operating system. Consider **T**, the universal tag space of all possible HTML document source tags:

$$\{\forall\, e_j \in \mathbf{E}\ \exists\ \mathbf{I} = \{i_1, i_2, \ldots i_{|\mathbf{I}|}\} \text{ s.t. } (\mathbf{I} \subseteq \mathbf{T}) \wedge (\text{unsupported } (\mathbf{I}, e_j))\}$$

That is, each browsing environment supports only a portion of the overall tag space. All others are unrecognized, or incompliant, in the targeted browsing environment. Consequently, the tags in **I** could be considered to be bug patterns for web pages rendered in environment e_j and cross-browser accessibility of a web site can be evaluated in a code review by detecting the presence of tags $i_n \in \mathbf{I}$.

Example 1:

The tag <div>, though a part of the comprehensive tag set **T** and supported by Internet Explorer, is unrecognizable in browsing environments that feature Netscape. Consequently, <div> would be an element of **I** for environments **E** where **B** in the corresponding triplet is Netscape.

Definition 2: Bug Pattern Completeness

The accuracy of code reviews based on the set of tags in **I** is largely dependent upon the accuracy and completeness of the description for **I**. If for instance:

$$\{\exists\ t_i \in \mathbf{T} \text{ s.t. } (t_i \notin \mathbf{I}) \wedge (\text{unsupported}(t_i, e_j))\}$$

performance of a static analyzer that does not include t_j as a bug pattern is compromised.

Example 2:

Consider <blink>, a tag that is unsupported in browsing environments featuring Internet Explorer. If compliance evaluation was executed for a web page that incorporated the <blink> tag yet the tag was not listed in **I** as a bug pattern, the accuracy of the resulting report would be compromised, and developers would be subject to latent failures and a false confidence in compliance.

Given the usefulness of bug patterns in evaluating environment compliance and the need for comprehensive pattern sets, it is our position that developing tools which identify the presence of environment specific bug patterns in web pages and support mechanisms for updating and refining knowledge of tag compliance rules is a first step in improving web quality for users when browsing environment influenced accessibility issues are an issue.

4. RESEARCH DIRECTIONS

In an effort to address the accessibility challenges influenced collectively by the browser, browser version, and operating system used for web navigation, we have developed a tool that estimates web page interoperability based on knowledge of the HTML tags that comprise web pages and knowledge of browsing environment specific tag support. As a result, one research direction inherent in our approach is discovering efficient ways of comparing web page source code to a list of bug patterns or, in this case, unsupported HTML tags. These lists can be relatively long for each browsing environment. Determining efficient ways of representing the rules for each environment and conducting subsequent comparisons efficiently is fundamental to the success of any tool featuring this approach.

As noted before, the strength of a code review-based evaluation strategy is heavily dependent upon the comprehensive nature of tag support knowledge for various browsing environments. In other words, conducting code reviews with truncated or inaccurate knowledge can severely inhibit accurate compliance analysis, causing a page to appear as if it will render properly in an incompliant environment. Consequently, another research direction would be to identify effective ways of updating tag support knowledge as a means of ensuring that all incompliant tags are represented in the bug pattern set associated with a given browsing environment. One method of optimizing knowledge of tag support criteria is to design a learning mechanism that can estimate the compliance of a novel HTML tag, or a tag for which compliance or incompliance knowledge has not been established, based on observations of positive (accessible) and negative (inaccessible) examples of page presentation and functionality. The underlying theory of such a technique is that observation of HTML tags that are positively correlated with inaccessible web pages can provide insight into the root causes of inaccessibility.

The overall endeavor to detect browsing environment accessibility barriers and to ensure comprehensive knowledge of tag support fall under general research endeavors associated with bug isolation [4] and bug patterns[5]. In this case, bug isolation is a related endeavor because the attempt to discover new tag support rules is largely an endeavor to isolate faulty tags given observation of source documents. Bug patterns, of course, are related because faulty tags can be recast as bug patterns and used during a code review to detect possible accessibility barriers. Research directions in which measures derived from fundamental principles of either of these two concepts are also viable.

5. CONCLUSION

Enabling consistent rendering of web pages across heterogeneous browsing environments is one facet of a general endeavor to support and improve the quality of user experiences on the web. From our prospective, providing tools capable of detecting browsing environment accessibility issues is extremely important for supporting general web quality and there are numerous research directions entailed in developing effective solutions.

6. REFERENCES

[1] http://benoitbisson.com/blog/archives/000180.html#000180

[2] http://www33.brinkster.com/articlesupport/browser.html

[3] Hull L (2004) Accessibility: It’s not just for disabilities any more, Interactions **11**(2): 36-4

[4] Liblit B, Aiken A, Zheng A, Jordan M (2003) Bug Isolation via Remote Program Sampling, Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation: 141-154

[5] Allen E., Bug Patterns: An Introduction: Available at: htttp:///www-106.ibm.com/developersworks/library/j-diag1.html(2001)

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.