Feature-based Web Accessibility Testing for Disabled Users

December 8, 2008

Abstract

Web accessibility evaluation tools can significantly reduce the time and effort required to carry out website accessibility evaluations. When used carefully throughout the design, implementation, and maintenance phases of Web development, these tools can assist their users in preventing accessibility difficulties, repairing them, and improving the overall quality of Web sites. However, the analysis of existing tool capabilities and the feedback from disabled people has identified the necessity of having the automatic tool that determines the accessibility of web sites and can predict the foreseen accessibility barriers. There are a number of tools and tests that developers can use to validate their web sites but they do not guarantee that pages are accessible, they can only assist in designing them. In this proposal, we evaluate and compare the existing tools, group them by validation coverage area, and propose the guideline to developers on which tool can be used to check different levels of accessibility. Out of these groups, the best tool is picked and they are integrated as a plug-in into the existing framework [7]. Also we propose a novel approach to web pages accessibility testing based on machine learning techniques using the generated report from this framework as training features. It allows our tool to be easily adapted to a wide range of disability groups, or even to individual preferences. In addition, our tool could be used to predict the severity accessibility issues, which could also be based on the preferences of a particular population or indivdual.

1 Introduction

The rapid, progressive diffusion of Web sites and applications in several productive contexts of our modern society is laying the foundations of a renewed scenario of its development. The majority of websites have not been designed to meet the fundamental needs of groups of people challenged by impairments related to vision, hearing, motor skills and cognitive abilities. However, it is essential that all information and services the Internet offers are accessible by every person, regardless of health conditions or hardware/software limitations. Although there are web accessibility evaluation and validation tools available, they do not guarantee that web sites are accessible. These tools are good as a guide, a way of identifying problems with your code but there is a need for an efficient, systematic evaluation process that tests website accessibility and helps designing sites already accessible. Currently, there is no universal automatic tool that a developer can use while he is in the pro-

cess of designing his web site.

In this proposal, the problem of accessibility in web sites is addressed by comparing the existing tools, grouping them by validation coverage area, picking a tool out of these groups that capture the most inaccessibility issues and plugging them into the existing framework [7] that gives the report of all found inaccenssibilities. Further, based on this report, machine learning techniques are used to predict the inaccessible barriers and ability to accurately assign a severity to each discovered barrier.

2 State of the Art

The Net is becoming a universal medium, the conduit for most of the information that flows through peoples eyes and ears and into their minds. The advantages of having immediate access to such an incredibly rich store of information are many. The World Wide Web has become a powerful tool in our society, and every individual should share the right of gaining access to it. There are a few studies that identify the issue of providing web accessibility to individuals with cognitive deficits [15], while others identify the issue of accessibility to visually impaired users [5], [13], [14], [16], [17], [18].

An inaccessibility issue for individuals with cognitive deficits is difficulty of making choices and browsing from page to page while interacting with web content. Among other things, a way of providing visual structure to a page is very helpful for people with developmental disorders and/or autism. Currently, it is hard to determine automatically the level of navigation difficulty in a website.

One of the inaccessible issues for blind people who use a screen reader that reads the content of a webpage aloud to them is the presence of an "alt" tag for all IMG, INPUT, and APPLET elements to describe its function. While it is easy to determine whether an "alt" attribute is missing, it is quite difficult to automatically determine whether the contest is meaningful or clearly describes the function of an element. Such cases make a website inaccessible to blind users.

And the reality is: there are many people who have problems accessing information on websites because of their disabilities. Although this situation is far from being satisfactorily solved, some advances have been reached in the last few years. Some guidelines have been created:

- Section 508 of the Rehabilitation Act requires all federal agencies to follow rules to provide fully accessible websites (http://www.section508.gov).

- The recommendations of the W3C Web Accessibility Initiative3 (WAI/ W3C4 consortium) have accepted the essential requirements for Web content to be accessible to people with disabilities.

Web content accessibility helps people with disabilities access Web pages directly or use assistive technologies. Many people with visual disabilities have to rely on specialized software or hardware to access the Web. Strain et. al. [16] focus on a problem of accessibility testing for blind people who use a software package called a screen reader to read all the content on the Web page aloud to them. Takagi et. al. [17] analyze the navigability of web applications for blind users with a modified browser that access the web page orally. Regardless of the solution favored by the users with disabilities, if the content of the Web page is not available to their

remaining sensory channels then the page is not accessible to them.

Despite the significant effort of government and other agencies to provide guidelines for making websites accessible, neither websites [18] nor tools that check web pages for accessibility for users with disabilities are satisfactory [11]. Previous research focused mainly on two different aspects of testing websites to evaluate their accessibility: testing accessibility of websites and tools that test accessibility of web pages.

2.1 Testing Accessibility of Websites

Testing accessibility of websites compares accessibility of web pages and search engines for blind and sighted users [5], [14] and offers methods for accessibility testing [1], [3], [13]. Buzzi et. al. [5] evaluated seven search engines. The results showed that accessibility is greatly neglected. Out of seven search tools analysed, only Google conformed to level A of accessibility as specified by WCAG 1.1, assuring a minimal level of accessibility. Petrie et. al. [14] investigated the accessibility of websites, noticing that visually impaired and non-disabled people often encounter the same problems but these problems appear to be amplified for people with blindness. Law et. al. [12], developed a process (SERPA) that gives guidelines to programmers to balance the volume and scope of fixes needed to improve site accessibility.

Another key ingredient of testing accessibility of websites is the sampling method to reduce the number of pages to be tested. Brajnik et.al [3] used various sampling methods to find out how they affect accessibility testing. He reported that the accuracy of the methods depends on the metric that is being used (conformance metric, WAQM, and UMEM are being considered there). This means that the choice of sampling method should be done after careful analysis. He found out that the conformance metric is by far the most sensible. Brajnik et. al. [1] and Mankoff et. al. [13] compared a diverse range of accessibility testing methods to determine which are most effective, and compared their relative strengths and weaknesses. The results show that no single evaluator or tool could be counted on to find a high percentage of accessibility problems of any type (WCAG or Empirical). However, evaluators using a combination of a screen reader and monitor were most consistently effective at finding both empirical and WCAG accessibility problems.

2.2 Accessibility Testing Tools

Validation tools that can perform automated and semi-automated accessibility testing of web pages are becoming more popular. While various tools, such as Bobby, LIFT, and WAVE are able to verify the compliance of a Web site against the checkpoints of the WCAG 1.0, they do not fix and therefore do not guarantee that the web site is accessible. Kirchner et.al. [10] tested and compared the capabilities of nine current accessibility testing tools. While testing scripts and/or CSS in the page, most of the tools are not evaluating them and some give misleading results. Every tool out of these nine inspected tools focus on different aspects of the accessibility problem and can test only up to five pages at a time. Di Lucca et.al [6] developed a tool that supports accessibility analysis and fixes the identified problems by applying the proposed fixing rules to the source code of the pages. The explorative study has been carried out on accessibility problems deriving from the use of client scrips on 20 pages

from 10 Internet Web sites. It demonstrated the effectiveness of model, process, and tool in the identification of accessibility violations. All of the websites except two had pages with potential accessibility issues and those two did not have violations only because they did not use scripts.

Feigenbaum et. al. [8] developed a semi-automated tool, RAVEN, for inspecting and validating Java rich-client GUIs for accessibility and introduced an approach to building rule-based validation and inspection tools for software products. Brajnik et. al. [2] also developed a semi-automated tool, SAMBA, to merge human judgments with automatic testing that provides values to understand how accessible a web site is with respect to some specific user category. Freire et. al. [9] provided the evaluation of dynamic Web pages since the process of verification of dynamic Web pages has more than one possible output for each different input. The very recent work is to integrate accessibility testing tools into a framework that in combination gives good coverage [7]. Unfortunately, there is no evaluation of any of these semi-automatic and automatic tools.

2.3 Limitations/Unaddressed Problems

There are numerous unaddressed problems in the website accessibility for disabled users. The most important is that there are no tools evaluating dynamic pages. The only one tool, (RAVEN), developed by IBM is inspecting and validating Java rich-client GUIs for accessibility. However, there is no evaluation of this tool.

Another limitation is that there are a lot of tools that identify problems in static HTML but, first, most of them are far away of being satisfactory. Second, there are so many of them that a developer just doesn't have time to test his website using all of them.

Another limitation is that there is no a universal automatic tool for checking website accessibility.

3 Challenges and Goals

Many tools have been developed to identify and/or fix accessibility problems in websites. A list of these tools can be found on the Web Accessibility Initiative (WAI) website. They are divided in three groups: evaluation tools that perform a static analysis of pages and generate reports of found accessibility problems, repair tools that help designers to make their pages accessible, and filter and transform tools that supply disabled users with assistive technologies. However, none of them are completely satisfying. Every tool is focused on different aspect of the accessibility problems and a designer has to know which tool is worth to use and has to apply different tools to have his site fully accessible. In this proposal, we would like to compare existing tools, similar to how it was done in [10] and group them by different levels of accessibility.

There is a group of people from IBM that implemented a system that integrates testing tools as plug-ins and gives a report with accessibility problem findings. There was no evaluation of this automatic tool done. It would be useful to pick tools to plug-in into the framework based on some criteria.

Another open key issue is to have a predicting tool that can foresee the severity of accessibility barriers. It would be useful to create accessibility models based on different sets of users or on a particular users preferences. This would be valuable simply because what could be a prob-

lem for one user or disability group may or may not be an issue for another group. The proposed tool would provide a single framework that could easily be adapted to a wide range of populations. Also it would be useful to provide explainations for the results of our model. For example, if a particular web page is graded poorly by the model, the tools output should indicate the features that contributed the most toward the rating. This gives developers an indication of where problems exist. Also the tool generates a list of most influential features. Such lists could be customized for a specific individual or population.

4 Proposed Research

4.1 Tool Comparison

We propose to group the tools by specific areas and highlight the most effective tool out of each group in terms of accuracy, precision, ease of use, and efficiency to simplify a designer's work. That is the designer does not need to test his website with, say, 10 tools that detect the same problems in the site. Instead he can use only one tool for each addressed violation.

4.2 Tool Integration

Previous work presented a plug-in framework for integrating multiple accessibility tools [7]. However, this paper provides limited empirical evaluation of how well the proposed framework is. We propose to integrate the most efficient tools in this framework to have the best coverage of accessibility violations.

4.3 Machine Learning Model

Currently, a predicting tool does not exist. We propose to apply machine learning techniques to predict barriers to accessibility, based on features of the accessibility violations report from the framework described above. This method could also be used to predict the severity of the barriers. A model would be generated using the Weka toolkit.

4.3.1 Feature Identification

The first step of our research would be to select a set of features to be used by the model. Here is a list of potential features that could be used:

- Number of images with missing ALT tag / Total number of images.
- Total number of images.
- Number of HTML validation errors.
- Number of frames with missing title / Total number of frames.
- TABLE elements mis-used for formatting, instead of data.
- Document length (in characters).
- Fleisch-Kincaid grade level.

This example list would be expanded upon for the actual research. The final features would be based on a study of previously identified barriers to accessibility, as well as the authors' intuition as to which features could best predict barriers to accessibility.

Note that the last feature listed here is actually the output of an existing natural language readability metric. This approach could also be

used to integrate the output of existing automated tools into our framework.

4.3.2 Model Creation

We use machine learning techniques to build a classification model. A benefit of this approach is that we do not need to know ahead of time which features will be useful to a particular population, nor do we need to know the severity of issues associated with each feature. Another benefit is that if any complex interactions between features exist, there are existing machine learning classifiers that account for this interactions [4].

A portion of the data collected from the integrated framework report would then be used to train a model using one of the classifers from the Weka framework. The trained model would match features extracted from a web page with the expected accessibility rating of that page. The severity of the barriers could be graded on a simple numeric scale (for example, 1 being "slightly inaccessible", and 5 being "completely inaccessible").

4.3.3 Tool Development

Our implementation will provide two distinct interfaces, both written in Java. A GUI interface could improve tool usability, but is not required. The two interfaces are:

- *A model training interface*: This interace is provided so that a model can be trained on data from new populations. The input for this interface is a set of HTML documents each labeled with one or more human accessibility ratings.
- *A testing interface*: This interace allows users to test the accessibility of a web page based on a particular model. There is also an option to load multiple models, in order to test accessibility for various populations at the same time.

5 Evaluation Plan

To give the guidelines to designers, the evaluation of all existing tools would be based on accuracy, precision, ease of use, and efficiency.

To evaluate the framework, an empirical study would evaluate the effectiveness of this approach as compared to individual tools.

To evaluate the predicting tool, the model generated by the accessibility violations report from the framework will be compared to an evaluation of the website by actual users from several different disability groups. Since this data came from actual users, we assume that it represents the real-world barriers to accessibility experienced by the respective disability groups.

For comparison, we will also evaluate the performance of existing automated tools. If an existing tool is targeted toward a particular disability, then the comparison will focus only on test data from individuals with that disability. We would expect the proposed method to perform favorably when compared with these existing tools, especially in terms of accurately predicting the severity of the barriers.

Next the proposed method would be compared to a technique known as expert review. For this technique, web developers–armed with accessibility guidelines and special interfaces such as a screen reader–try to locate barriers to accessibility.

6 Summary of Foreseen Contributions

In summary, the main contributions to the state-of-the-art are:

- We evaluate, compare the existing tools, and group them together by validation area.
- The best tool out of each group can be integrated in existing plug-in framework done by [7]. We propose to carry out the evaluation of a report with accessibility issues given by this framework.
- We expect to contribute an accessibility testing method that is easily adapted to new populations or individuals. This method would also provide an empirically-based accessibility rating, based on a 5 point numeric scale. We would also provide an explanation for the rating, indicating which features contributed most heavily to the final rating. The rating and explanation together could be used for planning and prioritization of remediation tasks. We would also enumerate to what degree each feature contributes to the final rating. From this we could infer which features are most (and least) important to a particular population.

References

[1] Brajnik, G. Automatic testing, page sampling and measuring web accessibility, March 2008. `http://users.dimi.uniud.it/ giorgio.brajnik/papers/csun08.pdf`.

[2] Brajnik, G., and Lomuscio, R. Samba: a semi-automatic method for measuring barriers of accessibility. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility* (New York, NY, USA, 2007), ACM, pp. 43–50.

[3] Brajnik, G., Mulas, A., and Pitton, C. Effects of sampling methods on web accessibility evaluations. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility* (New York, NY, USA, 2007), ACM, pp. 59–66.

[4] Buse, R. P., and Weimer, W. R. A metric for software readability. In *ISSTA '08: Proceedings of the 2008 international symposium on Software testing and analysis* (New York, NY, USA, 2008), ACM, pp. 121–130.

[5] Buzzi, M., Andronico, P., and Leporini, B. Accessibility and usability of search engine interfaces: Preliminary testing. In *Adjunct Proc. of 8th ERCIM UI4ALL Workshop* (2004).

[6] Di Lucca, G., Fasolino, A., and Tramontana, P. Web site accessibility: identifying and fixing accessibility problems in client page code. *Web Site Evolution, 2005. (WSE 2005). Seventh IEEE International Symposium on* (Sept. 2005), 71–78.

[7] Englefield, P., Paddison, C., Tibbits, M., and Damani, I. A proposed architecture for integrating accessibility test tools, August 2005. `http://www.research.ibm.com/journal/sj/443/englefield.html`.

[8] Feigenbaum, B., and Squillace, M. Accessibility validation with raven. In *WoSQ '06: Proceedings of the 2006 international workshop on Software quality* (New York, NY, USA, 2006), ACM, pp. 27–32.

[9] Freire, A. P., and de Mattos Fortes, R. P. Automatic accessibility evaluation of dynamic web pages generated through xslt. In *W4A '05: Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A)* (New York, NY, USA, 2005), ACM, pp. 81–84.

[10] Kirchner, M. Evaluation, repair, and transformation of web pages for web content accessibility. review of some available tools. In *WSE '02: Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02)* (Washington, DC, USA, 2002), IEEE Computer Society, p. 65.

[11] Kirchner, M. Benchmark for testing the evaluation tools for web pages accessibility. *Web Site Evolution, 2003. Theme: Architecture. Proceedings. Fifth IEEE International Workshop on* (Sept. 2003), 66–73.

[12] Law, C., Jacko, J., and Edwards, P. Programmer-focused website accessibility evaluations. In *Assets '05: Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility* (New York, NY, USA, 2005), ACM, pp. 20–27.

[13] Mankoff, J., Fait, H., and Tran, T. Is your web page accessible?: a comparative study of methods for assessing web page accessibility for the blind. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2005), ACM, pp. 41–50.

[14] Petrie, H., and Kheir, O. The relationship between accessibility and usability of websites. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), ACM, pp. 397–406.

[15] Sevilla, J., Herrera, G., Martínez, B., and Alcantud, F. Web accessibility for individuals with cognitive deficits: A comparative study between an existing commercial web and its cognitively accessible equivalent. *ACM Trans. Comput.-Hum. Interact. 14*, 3 (2007), 12.

[16] Strain, P., Shaikh, A. D., and Boardman, R. Thinking but not seeing: think-aloud for non-sighted users. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2007), ACM, pp. 1851–1856.

[17] Takagi, H., Saito, S., Fukuda, K., and Asakawa, C. Analysis of navigability of web applications for improving blind usability. *ACM Trans. Comput.-Hum. Interact. 14*, 3 (2007), 13.

[18] Zeng, X., and Parmanto, B. Web content accessibility of consumer health information web sites for people with disabilities: A cross sectional evaluation, 2004. `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1550595`.