An Educational Tool for Generating Inaccessible Page Examples Based on WCAG 2.0 Failures

Atheer S. Al-Khalifa
Electronic and Computer Institute
King Abdulaziz City of Science and Technology
Riyadh, Saudi Arabia
aalkhalifa@kacst.edu.sa

Hend S. Al-Khalifa
Information Technology Department
King Saud University
Riyadh, Saudi Arabia
hend@ccis.edu.sa

ABSTRACT

One of the problems encountered while teaching web accessibility evaluation to undergraduate students is the lack of proper educational tools that support learning accessibility barriers modularly.

This paper presents an online educational tool called Accessibility Example Generator (AEG), designed to assist instructors in the process of creating examples of inaccessible web pages that violate the accessibility guidelines of WCAG 2.0. The online tool supports generating the examples in a form which can be received and reviewed easily by undergraduate computing students. By using a sequence of tailored checks, the instructor can choose which failure or combination of failures the example should include.

Utilizing such a tool while teaching web accessibility will enrich the learning and understanding of WCAG 2.0 guidelines through generating modular examples that will not overwhelm the student and at the same time will help spread the knowledge of accessibility through future developers.

Categories and Subject Descriptors

H.5.2 [Information Interface and presentation]: User Interfaces – Evaluation/methodology. K.4.2 [Computer and Society]: Social Issues –Assistive Technologies for persons with disabilities.

General Terms

Measurement, Human Factors, Verification.

Keywords

Web Accessibility Evaluation, Web Accessibility Guidelines, WCAG 2.0, Educational Tool.

1. INTRODUCTION

Web Content Accessibility Guidelines (WCAG) published by the World Wide Web Consortium (W3C) is considered one of the most accepted guidelines for web accessibility. The guidelines explain how to make the web accessible to people regardless of their disabilities and targeting in particular the information in a web page or web application, including text, images, etc.

In December 2008 the W3C have published the second version of WCAG known as WCAG 2.0, aimed to be technology-independent and testable [7]. The structure of WCAG 2.0 separate between user requirements and actual implementation details. User requirements are classified into three layers of guidance: Principles, Guidelines and Success Criteria [7]. Besides, there are additional layers of guidance that supplements WCAG 2.0 provided by an external document called Techniques. A technique is the part where the level of implementation details is located; it provides methods for addressing the successes criteria.

There are three types of techniques: Sufficient Technique, Advisory Technique and Common Failures [7]. In this paper we are interested in the common failures, which are examples of bad practices that caused web pages to fail in meeting the success criteria. These failures can be used as a first step to teach the essence of web accessibility to undergraduate students.

Therefore, this paper presents a tool that assists instructors in producing modular and easy-to-evaluate examples for students based on selected WCAG 2.0 common failures. The tool will guide the instructor through a sequence of checks required for generating the designated example. The example is then given to the student for accessibility evaluation.

2. MOTIVATION

Learning web accessibility and implementing it can be a hard task for beginners. It cannot be done only by reading the guidelines and following them in the development process. Instead students or novice web developers need to practice evaluating and testing web pages in order to gain more understanding of them.

To justify this claim, Alonso et al. [3] have run an experiment focusing on the testability of WCAG 2.0 by undergraduate students in an intensive course of web accessibility. All students were asked to manually evaluate the accessibility of the same web page based on the 25 level-A success criteria. The results showed that WCAG 2.0 is far from testable by students according to three causes; one of them is related to the students' understandability of the guidelines. So Alonso et al. have concluded that with better training students can easily improve their results [3].

With this problem in mind, our proposed solution will contribute in providing the required training to the students by the generation of modular accessibility barriers examples based on WCAG 2.0 common failures. In other words, the need to implement a tool that helps undergraduate students or novice web developers in understanding the guidelines of web accessibility is facilitated by training them on small examples in order to improve their skills in evaluating web accessibility.

3. RELATED WORK

Since publishing WCAG, several tools have been released to assist developers in evaluating the accessibility of web pages based on its guidelines [5]. Few of them were designed to facilitate and help novice developers or undergraduate students in the process of learning the essence of web accessibility evaluation.

One of these tools is the Accessibility Evaluation Assistance (AEA) tool proposed by Bailey and Pearson [4]. AEA is an educational tool to support the accessibility evaluation process. It is not considered an automated evaluation tool per se; instead it requires an auditor to manually examine the web page by selecting the features of the website (s)he wishes to evaluate through tailored checks based on a specified user group, or content features of the website. Then the tool will priorities the relevant checks (tests) for the auditor, which will help in drawing more accurate conclusions about the accessibility of individual websites and facilitating the prioritization of fixes [4].

In contrast, Abou-Zahra and Cooper [1] presented a Test Sample Repository containing minimal web content examples that demonstrate accessible and inaccessible implementation of the WCAG 2.0 techniques. The repository depends on the collaboration of the public to contribute in enriching its content especially developers of web accessibility evaluation tools [1]. However, the repository is still incomplete and it is currently being actively developed by the W3C Web Accessibility Initiative (WAI).

Finally, it is worth mentioning that the W3C has a very decent multi-page resource suite showcasing practical examples on accessibility barriers, called the "Before and After Demonstration" [2]. Yet, these examples contain many accessibility barriers that might overwhelm early accessibility learners.

Therefore, our Accessibility Example Generator (AEG) will benefit from the previous works such that it is similar in concept to the AEA tool in terms of using it as an educational tool; and it also takes the advantage of having a repository of examples akin to the W3C repository.

4. THE EXAMPLE GENERATOR TOOL

Our proposed tool is designed as an educational support tool to help university instructors in generating simple examples on common failures of WCAG 2.0 success criteria. The philosophy behind the tool can be considered as a genre of problem-based learning, where students solve problems and reflect on their experiences.

The tool creates materials for a human inspector to manually evaluate. It aims to use existing resources i.e. WCAG 2.0 common failures, which will direct the process of systematically producing accessibility examples for the instructor.

The tool includes a failure repository, which holds the code snippets of examples. As a first step the failure repository is filled with failure examples found in the W3C "Failures for WCAG 2.0" web page [6]. The schema for the database includes:

- Feature ID - Identification number of a failure pattern.
- Failure ID – Identification number of the common failure.
- Description – Elaborate description of the failure example.
- HTML – The code snippets that should be included in the Body part of the code.
- CSS - The code snippets that should be included in the CSS part of the code.
- JS - The code snippets that should be included in the JavaScript part of the code.
- Server-side script - The code snippets that should be included in the Sever side Script, e.g. PHP, ASP.

The AEG tool provides a web interface that enables the instructor to generate examples easily. The tool will prompt the instructor to enter the features of the intended example by selecting them through contextual checkboxes, as shown in Figure 1. These features were extracted from the general patterns observed while analyzing the 89 WCAG 2.0 common failures.

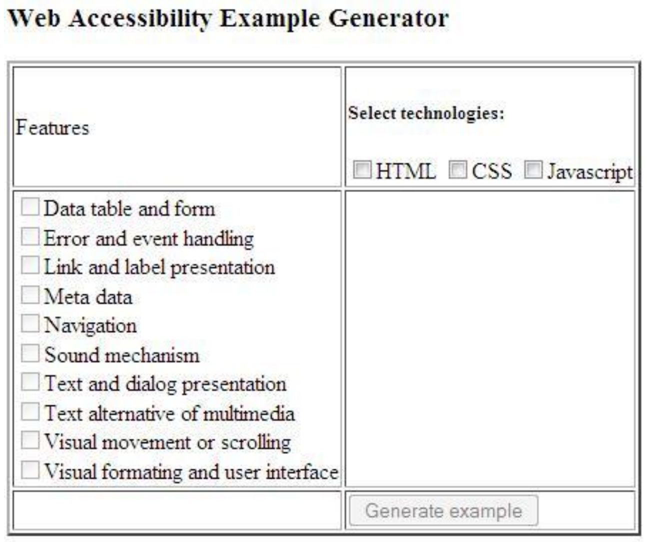
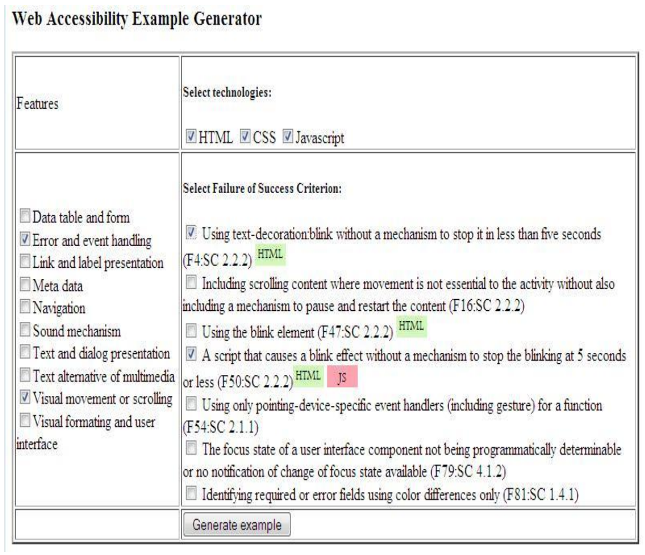

Web Accessibility Example Generator
Features
Select technologies:
HTML
CSS
Javascript
Data table and form
Error and event handling
Link and label presentation
Meta data
Navigation
Sound mechanism
Text and dialog presentation
Text alternative of multimedia
Visual movement or scrolling
Visual formating and user interface
Generate example

Figure 1 – AEG Tool Web Interface

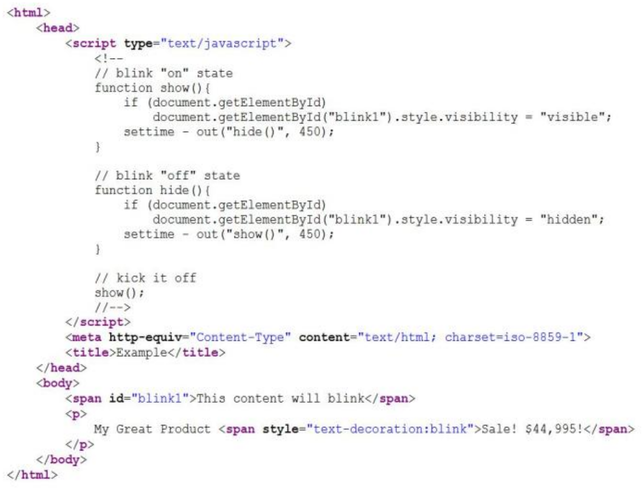
Once a feature or a set of features have been selected, the instructor is opted to select the type of technology that the failure occurs with. Then a list of failures associated with the technology is displayed. The instructor can then select one or more checkboxes of the associated failures, as shown in figure 2.


Web Accessibility Example Generator

Features	**Select technologies:** [✓ HTML] [✓ CSS] [✓ Javascript]
☐ Data table and form [✓ Error and event handling] ☐ Link and label presentation ☐ Meta data ☐ Navigation ☐ Sound mechanism ☐ Text and dialog presentation ☐ Text alternative of multimedia [✓ Visual movement or scrolling] ☐ Visual formating and user interface	**Select Failure of Success Criterion:** [✓ Using text-decoration:blink without a mechanism to stop it in less than five seconds (F4:SC 2.2.2)] HTML ☐ Including scrolling content where movement is not essential to the activity without also including a mechanism to pause and restart the content (F16:SC 2.2.2) ☐ Using the blink element (F47:SC 2.2.2) HTML [✓ A script that causes a blink effect without a mechanism to stop the blinking at 5 seconds or less (F50:SC 2.2.2)] HTML JS ☐ Using only pointing-device-specific event handlers (including gesture) for a function (F54:SC 2.1.1) ☐ The focus state of a user interface component not being programmatically determinable or no notification of change of focus state available (F79:SC 4.1.2) ☐ Identifying required or error fields using color differences only (F81:SC 1.4.1)
	Generate example

Figure 2: Related Failures of the selected Features

Next, by clicking on the "Generate example" button the tool will create an html file (or more if it is needed) by fetching the code snippets of the selected failure(s) from the repository and placing them in the right position inside the html file template (Figure 3). Finally a .zip file will be produced containing the file(s) ready to be inspected by the students.


```html
<html>
    <head>
        <script type="text/javascript">
            <!--
            // blink "on" state
            function show(){
                if (document.getElementById)
                    document.getElementById("blink1").style.visibility = "visible";
                settime - out("hide()", 450);
            }

            // blink "off" state
            function hide(){
                if (document.getElementById)
                    document.getElementById("blink1").style.visibility = "hidden";
                settime - out("show()", 450);
            }

            // kick it off
            show();
            //-->
        </script>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <title>Example</title>
    </head>
    <body>
        <span id="blink1">This content will blink</span>
        <p>
            My Great Product <span style="text-decoration:blink">Sale! $44,995!</span>
        </p>
    </body>
</html>
```

Figure 3: The final generated HTML file that contains the accessibility problems

To gain more understanding of the tool, the following scenario explains step by step the process of generating an example:

1. Let us assume that the instructor wants to generate an example that has accessibility failures in the following features: "Error and event Handling" and "Visual movement and scrolling" related to HTML, CSS, and JavaScript technologies.
2. The instructor will check the required technologies then the related features will be activated. Once the instructor chooses the required features, the tool will dynamically fetch the related failures from the database based on the Features' IDs.
3. Upon retrieval of features' failures, the tool will create a checkbox containing each failure description along with a small icon beside it showing which technologies associated to this specific failure.
4. Next, the instructor can choose the desirable failures and then click on "Generate Example" button to create an HTML file(s) containing code snippets of selected failures. Technically speaking, an HTML file is created by appending a header opening tag and checking whether the selected failures have any code snippets in the JavaScript or CSS fields. If so, it will append a JavaScript opening tag then the code snippets and a closing JavaScript tag. The same process will go for the CSS code. The tool will then append a body tag and append any code snippets found in the HTML field.
5. Finally, the generated HTML file(s) are zipped and downloaded to the instructor's computer, so (s)he can send the example to the students for manual evaluation.

5. CONCLUSION

Teaching web accessibility to undergraduate students requires exposing them to practical and easy to assimilate examples in order to guide them in the process of web accessibility evaluation. In this paper we presented AEG, an educational tool that helps instructors create accessibility failure examples based on WCAG 2.0 common failures. The rationale behind the tool is derived from the need to present manageable, modular and easy to inspect examples to novice accessibility learners. This will help in overcoming the problem of finding proper examples that tackles specific accessibility barriers when presented to students.

Since AEG tool is currently capable of producing failure examples based on WCAG 2.0 common failures, which are limited in number; yet, a future extension of this tool will include merging it with a dedicated (i.e. form based) authoring tool to create more examples in order to populate the current failure repository. The extension will work as a collaborative tool to enable instructors around the world to build up the repository with their expertise. Moreover, the integration with WCAG 2.0 test samples repository, upon its completion, will enrich the tool with variety of examples.

Another further extension to this tool is to build an auto grader system to automatically grade the students based on their answers. This will give the students an immediate feedback on their learning process.

Another interesting improvement to the tool is to have it inject the failure snippets into examples provided by the instructor. This can be approached by highlighting a specific line of code, and then the tool identifies the tag and displays the possible associated failures.

However, since the tool is still in its beta release, an evaluation of it has yet to be conducted in the near future. As we plan to improve the tool, we will conduct extensive evaluations and present the results in future paper(s).

Finally, the idea of combining the AEG with AEA tool and similar futuristic educational tools in a complete suite of instructive tools dedicated for web accessibility will create a rich learning experience that will benefit both the instructor and the student.

6. REFERENCES

[1] Abou-zahra, S., & Cooper, M. (2009). WCAG 2.0 Test Samples Repository. Human-Computer Interaction - HCI (pp. 619-627). Berlin / Heidelberg: Springer.

[2] Before and After Demonstration. (n.d.). Retrieved December 25, 2010, from The World Wide Web Consortium (W3C): http://www.w3.org/WAI/EO/2005/Demo/

[3] Alonso F., Fuertes J., González A., and Martínez, L. On the testability of WCAG 2.0 for beginners. 2010 International Cross Disciplinary Conference on Web Accessibility (W4A). New York, USA: ACM New York.

[4] Bailey, C., & Pearson, E. (2010). An educational tool to support the accessibility evaluation process. 2010 International Cross Disciplinary Conference on Web Accessibility (W4A). New York, USA: ACM New York.

[5] Complete List of Web Accessibility Evaluation Tools. (n.d.). Retrieved December 25, 2010, from The World Wide Web Consortium (W3C): http://www.w3.org/WAI/ER/tools/complete

[6] Failures for WCAG 2.0. Retrieved December 25, 2010, from The World Wide Web Consortium (W3C): http://www.w3.org/TR/WCAG-TECHS/failures.html

[7] Web Content Accessibility Guidelines (WCAG) Overview. (n.d.). Retrieved December 25, 2010, from World Wide Web Consortium (W3C): http://www.w3.org/WAI/intro/wcag.php