Automated Accessibility Evaluation Software for Authenticated Environments

A Heuristic Usability Evaluation

Elisa Maria Pivetta, Daniela Satomi Saito, Carla da Silva Flor, Vania Ribas Ulbricht, and Tarcísio Vanzin

Post-graduate Program in Knowledge Engineering and Management, Federal University of Santa Catarina, Florianópolis, SC, Brazil
elisa@cafw.ufsm.br,
{daniela.saito,carla.flor,vrulbricht,tvanzin}@gmail.com

Abstract. Web accessibility has been the subject of much discussion regarding the need to make Web content accessible to all people, regardless of their abilities or disabilities. While some testing techniques require human intervention, accessibility can also be evaluated by automated tools. Automated evaluation tools are software programs that examine the code of Web pages to determine if they conform to a set of accessibility guidelines that are often based on the Web Content Accessibility Guidelines Version 2.0 (WCAG 2.0), developed by the World Wide Web Consortium (W3C). In this context, the purpose of this study is to analyze an automated software program for evaluating authenticated environments and verify the usability of this tool, since automated systems require precision and reliability in terms of both results and use in any type of environment. With this in mind, this paper aimed at evaluating the ASES software by means of a heuristic evaluation carried out by three experts. The analysis revealed major accessibility problems, as well as improper functioning of available tools and inconsistency of results. Furthermore, ASES was found to have problems of efficiency, interaction, validity, and reliability in the results presented. Considering that this is an open-source accessibility testing tool that can be found on a government web site, the correction or improvement of the system's deficiencies identified in this study is highly recommended, as there is a lack of software available to evaluate authenticated environments.

Keywords: Automated evaluation tool, heuristic evaluation, usability.

1 Introduction

Web accessibility has been the subject of much discussion regarding the need to make web content accessible to all people, regardless of their abilities or disabilities. According to the W3C [1], web accessibility refers to allowing all people to perceive, understand, navigate, interact with and contribute to the web.

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2014, Part IV, LNCS 8516, pp. 77–88, 2014.

In this context, accessibility evaluation encompasses a range of procedures aimed at detecting known accessibility problems, such as violations of guidelines, system failures, errors, or user performance indicators [2]. Several approaches have been used to assess the accessibility of a site. Automated evaluation tools, for example, are software programs that test virtual environments for accessibility by analyzing their code to determine how well they meet the guidelines established for the inspection.

According to [3], evaluation methods differ in terms of their validity, usefulness, reliability, and efficiency. With these considerations in mind, the purpose of this study was to analyze automated software designed to evaluate authenticated environments in order to verify its usability precision and reliability. The software selected for analysis is referred to as ASES (Site Accessibility Evaluator and Simulator) [4]. The criteria used for tool selection focused on the compliance with the WCAG 2.0, subscription to the open-source philosophy, and validation of an authenticated environment – in this case, Moodle [5]. In an initial survey, seven automated accessibility evaluation tools were identified, but only two were selected: WAVE and ASES.

The objective of the first phase of this study was to evaluate WAVE, and the results are described in [6]. The present paper reports on the second phase, which centered on the examination of the ASES software. The analysis was performed by three experts, as suggested by Nielsen [7], and consisted of conducting a heuristic usability test founded on the ergonomic criteria presented in [8]. Each of the experts analyzed the tool according to criteria based on a severity scale. After the individual evaluations were completed, each evaluator verified the arguments used by their colleagues in order for them to reach a consensus and assign a collective score. In doing so, errors found individually could be assessed and reassessed by all experts.

2 Automated Evaluation Software

Automated evaluation software, also known as validator, evaluator and online validator, scans the code of a web page and checks its content for accessibility. These tools help determine if a given interface was developed according to accessibility standards [9].

Some of the automated accessibility validation software tools available online for free meet the minimum requirement for selection – i.e., compliance with the recommendations of both WCAG 1.0 and WCAG 2.0. The initial plan was to choose from the pool of programs suggested by WAI, but the list was found to be outdated[1], as none of the tools singled out adhere to the standards for accessibility set forth in WCAG 2.0. Thus, we used the programs picked out by [10] in addition to other tools identified by the authors of the present work. Table 1 presents the selected tools.

[1] http://www.w3.org/WAI/ER/tools/complete

Table 1. Automated Evaluation Tools

Software	Description
Access Monitor	It generates an accessibility report and a summary of results with an evaluation score that sums up and measures the level of accessibility achieved [11]. WAI Conformance Levels: A, AA, AAA.
AChecker (Public)	The results are grouped into three categories: known, likely, and potential problems [12]. WAI Conformance Levels: A, AA, AAA.
ASES 2.0	It is a tool designed to support compliance with existing accessibility guidelines in government web sites. Conforms to the standards proposed in the WCAG 2.0 [9] and e-MAG 3.0 [13]. WAI Conformance Levels: A, AA, AAA.
TAW3	It presents three categories of accessibility violations: problems, warnings, and not reviewed [14]. WAI Conformance Levels: A, AA (AAA: commercial).
WAAT	Java application [16]. WAI Conformance Levels: A, AA, AAA.
WAVE	It offers four types of reports as follows: errors, features and alerts; reading order and structure of the page; page display in text-only format; and identification of page headings [17]. WAI Conformance Levels: A, AA, AAA.
Worldspace FireEyes	Add-on for the Firefox browser, it tests static and dynamic content [18]. WAI Conformance Levels: A, AA.

Among the tools listed in Table 1, Worldspace FireEyes [18] and TAW3 [14] were excluded from analysis for testing for accessibility in only two of the three levels of compliance – A and AA.. It is noteworthy that the success criteria adopted for each level are determined according to the degree of difficulty individuals with disabilities are likely to experience when accessing information on the web, as compared to other audiences. Thus, the the A, AA, and AAA levels are relevant to the verification.

The next step was to run the other software tools and evaluate them with a Moodle environment being used for the testing. Some of the programs failed the evaluation process, proving to be only suitable for use in areas in which access is open. Only two of the tools listed in Table 1 obtained successful results in authenticated environments: WAVE [17] and ASES [4].

Then, the research process was divided into two parts: the first phase consisted of a study of the WAVE tool, as presented in [6]; the second phase, described in this paper, centered on the examination of the ASES software. The analysis was found to be opportune, as ASES was designed to evaluate the accessibility of web pages and is offered for download on a government web site.

A search was conducted using the CAPES [2] database and the search engine Google.com to take stock of the state of the art in the area. The results obtained called for an evaluation of ASES, for the only existing work on the subject was conducted by [19] and consisted of a review of the Lattes Platform[3] using the ASES software. No studies providing an evaluation of the tool were found.

[2] `http://www.periodicos.capes.gov.br.ez47.periodicos.capes.gov.br/`

[3] A Plataforma Lattes integra bases de dados de currículos, grupos de pesquisa e instituições brasileiras. `http://lattes.cnpq.br/`

2.1 ASES Software

ASES, the Portuguese acronym for Site Accessibility Evaluator and Simulator, is a Web site designed to support the adoption of accessibility guidelines by government agencies. It is free software released under the GNU Lesser General Public License (LGPL), and its features are described both on the web site [4] and in the software manual. ASES offers the following functions:

- Accessibility Evaluator – examines the source code of web pages to determine their level of adherence to established practices presented in the WCAG 2.0 [9] and e-MAG 3.0 [13] documents;
- CSS Validator – checks whether the CSS code conforms to the accessibility standards devised by the W3C;
- HTML (4.01 and XHTML) Validator – inspects HTML/XHTML syntax to ensure it complies with the W3C specifications;
- Screen Reader Emulator – calculates the shortest amount of time a screen-reader user would take to reach a certain point of a Web page;
- Low Vision Simulator – simulates conditions or disabilities that impair vision, such as hyperopia, myopia, color blindness, cataracts, glaucoma, and retinopathy;
- Visual Descriptor - associates images with their textual equivalents;
- Systematic Analysis of Images – checks the lack of text-alternatives to the same image on several different pages;
- Alternative Content for Script Tags – provides alternative content for script tags;
- Alternative Content for Object Tags – provides alternative content for object tags;
- Label Placement – helps ensure all form controls have descriptive labels;
- Insert DocType – assigns DocTypes (headers at the top of HTML documents) to the desired pages;
- Form Filler – helps fill in the blanks on a form;
- Device-Dependent Event Handler – points out events that require the user to have any specific input device that is missing, such as “onMouseClick” and “onMouseOver”. It also allows users to edit events to call a JavaScript function;
- Redundant Links for Image Maps – provides redundant links to image maps.

The first part of the evaluation process consisted of installing the ASES software and verifying its functionality in order to, later, analyze the tool according to the guidelines and recommendations of the WAI [15].

3 Evaluating the ASES Software – Methodology

Heuristic evaluation was the usability inspection method used to evaluate ASES. In a heuristic evaluation, experts can either put their knowledge of typical user behavior to use throughout the evaluation process [20], [21] or examine specific aspects of interfaces. Heuristics act as mnemonics for the evaluator’s existing knowledge that can be applied to the analysis of usability and accessibility issues.

For this type of testing, [22] suggests using from 3 to 5 reviewers to achieve a favorable benefit-cost ratio. Thus, the selected software tools were evaluated by three experts: two of them completed both their Bachelor's and Master's degree in Computer Science, and the third graduated with a degree in Design and holds a Master of Science in Knowledge Engineering and Management. The three evaluators are doctoral candidates and participate in a research group on web accessibility.

In this context, the evaluation process was performed using the steps adapted from [20] and [21]:

1. Study Design – definition of the heuristics to be used and their form of evaluation;
2. Evaluation Script – a guide is prepared to take the experts through the process;
3. Evaluation Period – each expert spends 1-2 hours independently inspecting the product, using the heuristics as guidance;
4. Findings and Problem Identification – by using the heuristics as the evaluation framework, the aspects that are not in line with the established practices are detected and recorded as problems;
5. Analysis, Evaluation, and Reporting – once identified, the problems are rated according to their severity level, and experts propose solutions to be presented in a report.

4 Evaluation Process

This section is intended to describe the methodological steps in conducting the evaluation of the ASES software.

4.1 Study Design

In view of the steps described for the first part of the evaluation process, we developed a set of heuristics for the accessibility testing using the ergonomic criteria proposed by [8] as a reference. They are as follows:

1. Guidance – refers to the means available to advise, orient, inform, and guide the users throughout their interactions with a computer (messages, navigation, alarms, etc);
2. Workload – concerns all interface elements that play an important role in reducing the users' cognitive and perceptual load and increasing the dialogue efficiency;
3. Explicit control – concerns both the system processing of explicit user actions, and the control users have on the processing of their actions by the system;
4. Flexibility – refers to the means available to the users to edit or customize the interface according to their needs;
5. Error Management – refers to the means available to prevent or reduce errors and to recover from them when they occur;
6. Homogeneity/consistency – refers to the way interface design choices are maintained in similar contexts, and are different when applied to different contexts;

7. Significance of codes and names – refers to how clear and meaningful the codes and names are to the users of the system;
8. Compatibility – refers to the match between the system and the users' expectations and needs when performing a task. It also concerns the coherence between environments and between applications;
9. Help and Documentation – relates to the ease of access to the system. The easier a system is to use, the smaller is the need for help or documentation. When required, Help should be easily accessible;
10. System Status – concerns the provision of information about what is going on with the system and of appropriate feedback within reasonable time.

When evaluating the heuristics presented above, a severity scale was used by the evaluators to estimate each problem's severity and frequency of occurrence, as well as the level of difficulty in overcoming it [23]. By doing so, it was possible to determine the scale values shown in Table 1.

Table 2. Scale used for rating severity of problems identified

Value	Evaluation	Description
0	I don't agree that this is a usability problem at all	This value may be obtained from the evaluation of an expert about a problem pointed out by another expert
1	Cosmetic problem only	Need not be fixed unless extra time is available
2	Minor usability problem	Fixing this should be given low priority
3	Major usability problem	Important to fix, so should be given high priority
4	Usability catastrophe	It is imperative to fix this

Source: Adapted from [23].

4.2 Evaluation Script

The script is a description of the procedures to be adopted by each of the experts in the evaluation process. It contains the instructions to conduct the testing and details the steps 3 and 4 of the methodology, namely, evaluation period and findings and problem identification, respectively. Each of the experts participating in the evaluation process was provided with the following information and recommendations:

1. Introduction to the ASES software;
2. Overview of the evaluation objectives;
3. Definition of the evaluation context – Moodle was considered a convenient environment for being a platform, used in a public universities, in which access is authorized via user authentication;
4. Time required to complete the task;
5. Criteria for use – outline of the heuristics to be used in the evaluation;
6. Rating scale – severity levels to be assigned to each dimension evaluated.
7. Recommendations as to the conduction of the evaluation process:

(a) Evaluating all of the software tools available – Accessibility Evaluator (evaluation of guidelines, contrast and screen reader), Source-code Checker (CSS, HTML 4.01 and XHTML), Tag Analyzer (Image and Object), Editors (Doctype and Events);
(b) Inspecting each area of the software at least twice:
    (i) In the first inspection, evaluators should work separately and avoid interaction with one other, assigning scores and taking notes of observations about the problems and errors found. This procedure is crucial for ensuring independent and unbiased reviews;
    (ii) In the second inspection, experts come together to discuss their scores and share their findings. Following peer review, evaluators may assign a new score if they wish to do so.

4.3 Evaluation Period

Evaluations were supposed to be completed within the recommended time frame of 1–2 hours, but, due to implementation problems, this time restriction was removed and experts were allowed to take as much time as necessary to conduct their analyses. The tests were performed on four computers, with some variations in the configuration of the equipment, as specified below:

- Windows Seven, 32 bit, 2 Gb RAM, 10-inch screen monitor;
- Windows Seven Professional, 64 bit, 2Gb RAM, 13-inch screen monitor;
- Windows Seven, 64 bit, 8 Gb RAM, 14-inch screen monitor;
- Windows Seven Professional, 64 bit, 4Gb RAM, 14-inch screen monitor;

A virtual learning environment based on the Moodle platform was selected for testing the program, so as to verify the scope and quality of an evaluation conducted on a web site in which users are allowed access via user authentication. In order to do so, a password was required to provide access to Moodle and inform ASES about the internal address of the environment, i.e., the URL (Uniform Resource Locator) to be tested.

4.4 Findings and Problem Identification

Initially, the analyses were conducted individually by each expert, as described in the evaluation script. Subsequently, once problems had been identified and notes had been taken, the results were discussed by the group so that each participant could review the scores assigned to the tools. Final evaluations were obtained by calculating the mean of the individual scores and the ratings determined by consensus between the evaluators.

Table 3 presents the results regarding to Accessibility Evaluator tool. The table below shows the mean rates of severity assigned by the experts to each heuristic evaluated and the observations describe the results of their analyses.

Table 3. Mean ratings and observation by experts

Heuristics	Mean rate	Relevant observations by Experts
Guidance	3	- Moving between open windows is confusing. - Some items only work by double-clicking on them, rather than by single-clicking, which is the default method. - When a report is being generated, there is an option to view in detail the page evaluated. However, there is no clear indication of this possibility, which may prevent the user from accessing the window for a "detailed view of the evaluation."
Workload	4	- In the "Save As" dialog box, the default file type for documents is not a valid format. - Switching between the Default Source and Edit Source tabs makes correction difficult, especially when a large number of errors are found. Note: the errors of the WCAG are only visible in the Default Source tab.
Explicit Control	4	- It is not possible to verify if the error or warning of the WCAG was corrected after editing the code. - Each time the source code is edited, the system asks if changes are to be saved. The user should be able to save the changes only when editing is completed. - The user is not able to capture the colors of the Web site evaluated using the contrast checker. - No feedback is given regarding the action being executed.
Flexibility	3	- The system allows users to increase or decrease font size and change color contrast options, but the shortcut keys are reversed. - Users are not able to search through the source code. - It does not allow users to adjust the layouts for code visualization (Default Source and Edit Source).
Error management	3	- The error message shown when the Accessibility Evaluator was accessed ("unable to access page content") does not provide any explanation to prevent the error from happening or help the user correct it.
Homogeneity/consistency	4	- Inconsistency between the number of errors presented in the summary report and those found in the detailed view of each page, in which only the eMAG-related errors could be visualized. - The features available in "Edit Source" and "Default Source" are not consistent with the other software functions. - The available formats in the "Save" and "Save As..." functions are not consistent with the other software features. - The "Save PDF Report" option does not provide information about what type of report is to be saved, but in the version in the ASES format this information piece is provided by the system. - There is no standardization of window messages. - Inconsistency and differences as far as icons and messages are concerned – that is, sometimes they appear and at other times they do not.
Significance of codes and names	2	- Use of unclear and/or unexpressive terms and abbreviations. - "Open URL" in the CSS Validator does not specify whether the URL should be a CSS or HTML file.

Table 3. (*continued*)

Compatibility	4	- The tool did not provide a complete evaluation of the authenticated area. - In the detailed view of the accessibility evaluation of pages, the only visible tab is Edit Code, which does not show the errors to be corrected. In order to visualize and correct the errors, it is necessary to switch between the Default Code and Edit Code tabs, rendering the completion of the task counterproductive.
Help and documentation	3	- Not all of the tools are working as indicated in the manual. - The manual is incomplete.
System status	4	- The system crashes. - The accessibility evaluator does not respond when reports are being generated and no feedback is given, as if no function was being run. - The results presented refer to the first Web site evaluated. - The low vision simulator does not work and no message is displayed when clicking on it.

4.5 Analysis, Evaluation and Reporting

This section presents the analysis and evaluation of the features available in the ASES software. In general, basic accessibility problems were identified in the interactions with the program, mostly because ASES did not show much flexibility. The enter and tab keys did not work when replacing the mouse. In this regard, it should be noted that navigation and tool usage via keyboard shortcuts is a basic accessibility requirement for software items. The options of increasing and decreasing font size and adjusting the contrast of the screen allowed for greater flexibility, but these functions were not available within all software areas.

Even with these problems, the scores assigned by the experts were similar for most of the heuristics, as evidenced in Table 2. As far as the divergences are concerned, in order to reach a consensus, the heuristics were reviewed and a new evaluation of the problems detected was jointly conducted.

Regarding the final results obtained, the problems associated with workload, homogeneity/consistency, compatibility, and system status were considered critical and in need of urgent solution.

The workload was affected by the improper functioning of window error messages and source-code warnings, which did not show pertinent data, thereby impairing the efficiency of the inspection task performed by the user.

The lack of homogeneity and consistency was observed in multiple areas of the software. There was no standardization in the system in terms of icons and windows, and save options presented different formats even when they had the same function.

As for compatibility, ASES was not found to be fully supported in authenticated environments, which was one of the assumptions of this study. In certain situations, the tool produced the error report when the authenticated URL was inserted, but when the source code was analyzed, the user was recognized by the system the user as a visitor, rather than as an authenticated user.

The system status was affected by constant system crashes, forcing a reboot of the operating system. The Accessibility Evaluator tool showed different behaviors when used by the experts. With the first expert, the system tested one URL and then stopped working. With the second and third experts, the results of the attempts to evaluate other URLs were reported in the reviews of the first address tested. However, after changing computers, the third expert was only able to conduct the check once, and then the tool did not work anymore. All experts tried to perform their tasks by closing down and restarting the software. However, they were not successful in this strategy. Other issues such as lack of feedback and system malfunctions of the Low Vision Simulator were also pointed out.

Still concerning the status of the system, the Alternate Content for Script Tags evaluation tool ran smoothly, but provided no feedback, i.e., it did not show whether or not there were errors. In this situation, it was not possible to know if the lack of response means that no errors were found. Ideally, a message should be sent as feedback whenever possible. For instance, in the Label Placement tool, when the site evaluated was correct, messages such as "Congratulations" were generated by the system. Another observation is that this tool, unlike the others, did not show the original source code, only the editable code.

Although not rated at a level of severity as high as that assigned to the previous heuristics, issues related to guidance, error management, flexibility, and help and documentation were considered relevant problems that should be given high priority, requiring corrections to improve software usability.

The guidance framework for software structure and navigation between open windows and tools was considered confusing. Apart from that, some functions only worked by double-clicking and the system did not provide any visual indication to inform the user that another window could be accessed from a certain moment on.

Regarding error management, no messages appeared to indicate when the tool was unable to access certain content. In addition, experts also noted the lack of possibilities for correction and provision of solutions when problems occurred. In the CSS Validator, the Doctype element was not recognized as a CSS document, which is true. However, the system defined it as an error, and this is neither an accessibility problem nor an error on the page.

With respect to help and documentation, ASES presented a detailed but incomplete manual and the Redundant Links for Image Maps tool, which could not be found when using the software.

Difficulties related to explicit user control were reported when using the Contrast Checker, since this tool only allows users to enter the values referring to the background and foreground colors of the page to be tested, not allowing them to pick colors from web pages through the feature known as eyedropper. For this reason, users are forced to use a graphics software program and have it installed on their computer. Problems related to Save options, the checking of errors, and WCAG warnings also posed difficulties to tool management.

Finally, the Significance of Codes and Names was considered a minor problem. With respect to this heuristic, the CSS Validator had a confusing denomination for the

"Open URL" feature, as, although it was possible to insert any hyperlink in the form, the tool only evaluates CSS files.

5 Final Remarks

This study evaluated the usability of automated accessibility evaluation software for an authenticated virtual environment. Software programs in this category present different approaches, formats, features, and benefits, and some perform a number of tests while others underestimate the problems on a web site. The ideal choice for a tool depends on a set of functions and on how the evaluator defines the priorities of the site to be tested. In this sense, the following validation criteria adopted, this study aimed to test the ASES automated evaluation software based on WCAG 2.0.

As previously discussed, experts identified the problems associated with the Workload, Explicit Control, Homogeneity/Consistency, Compatibility, and System Status criteria as the most critical ones. The analysis showed that the tool is not easy to use and has several problems that hamper access for users. The instability of the tool was found to be an obstacle to its usability, and the frequent errors, with no possibility of correction, revealed an inconsistent system.

Regarding the Guidance, Error Management, Flexibility, and Help and Documentation criteria, the results point to the need of having correction as a high priority. On the other hand, the errors relating to the Significance of Codes and Names were regarded as a minor problem that should be given low priority.

When testing an authenticated environment, ASES not always performs the analysis starting from the authenticated page, which prevents the use of this tool for projects of this kind. Moreover, the program may give not-so-attentive users the impression that the testing was performed on the URL address provided, leading to false expectations about the results of the analysis. In order to better understand the indications of errors and warnings of ASES, it is important that the users have some knowledge of computing. Also, because the automated evaluation tools give more technical results, a bit of coding knowledge is required.

We suggest that this accessibility evaluation tool be adapted to conform to the guidelines developed by the WAI, considering its four basic principles – perception, operation, understandability, and robustness, so that error reports, warnings, and other items can be classified within these categories. Adjustments to the problems identified are also recommended, mainly due to the lack of software able to assess authenticated environments.

Based on the results of the evaluation of ASES, we conclude that the software is not stable and the functionality of its tools depends on the performance of the system in which they operate. Thus, in order for the results of accessibility evaluations performed by ASES to be reliable, the system's interface should be redesigned and adaptations should be made with a view to achieving greater usability.

Finally, this analysis was not intended to promote, support, or degrade the software listed in this paper; rather, it aimed at encouraging the development of more usable accessibility evaluation tools.

References

1. World Wide Web Consortium, http://www.w3.org/
2. Brajnik, G.: Beyond conformance: The role of accessibility evaluation methods. In: Hartmann, S., Zhou, X., Kirchberg, M. (eds.) WISE 2008. LNCS, vol. 5176, pp. 63–80. Springer, Heidelberg (2008)
3. Brajnik, G.: Web Accessibility Testing: when the method is the culprit. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 156–163. Springer, Heidelberg (2006)
4. ASES - Avaliador e Simulador para a Acessibilidade de Sítios, http://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG/ases-avaliador-e-simulador-de-acessibilidade-sitios
5. Moodle.org, http://www.moodle.org
6. Pivetta, E.M., Flor, C., Saito, D.S., Ulbricht, V.R.: Analysis of an automatic accessibility evaluator to validate a virtual and authenticated environment. International Journal of Advanced Computer Science and Applications 4, 15–22 (2013)
7. Nielsen, J.: Why You Only Need to Test with 5 Users (2000), http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/
8. LabUtil: Laboratório de utilizabilidade: critérios ergonômicos, http://www.labiutil.inf.ufsc.br/CriteriosErgonomicos/LabIUtil2003-Crit/100conduc.html
9. Web Content Accessibility Guidelines 2.0, http://www.w3.org/TR/WCAG20
10. Al-Khalifa, H.S., Al-Kanhal, M., Al-Nafisah, H., Al-soukaih, N., Al-hussain, E., Al-onzi, M.: A Pilot Study for Evaluating Arabic Websites Usign Automated WCAG 2.0 Evaluation Tools. In: Proceedings of the 2011 International Conference on Innovations In Information Technology. IEEE Computer Society, Saudi Arabia (2011)
11. UMIC - Agência para a Sociedade do Conhecimento, http://www.acessibilidade.gov.pt/accessmonitor/
12. AChecker, http://achecker.ca/checker/index.php
13. e-MAG – Modelo de Acessibilidade de Governo Eletrônico – versão 3.0 (2011), http://www.governoeletronico.gov.br/biblioteca/arquivos/e-mag-3.0/download
14. TAW, http://www.tawdis.net/
15. Web Accessibility Initiative, http://www.w3.org/WAI/
16. Web Accessibility Assessment Tool, http://www.accessible-eu.org/
17. Web Accessibility Evaluation Tool, http://wave.webaim.org
18. Worldspace FireEyes, http://www.deque.com/products/worldspace-fireeyes
19. Barbosa, G.A.R., Santos, N.S.S., Reis, S.S., Prates, R.O.: Relatório da Avaliação de Acessibilidade da Plataforma Lattes do CNPq sob a Perspectiva de Deficientes Visuais. In: Anais Estendidos do IHC 2010, Belo Horizonte, Minas Gerais, pp. 139–150 (2010)
20. Paddison, C., Englefield, P.: Applying heuristics to accessibility inspections. Interacting with Computers 16, 507–521 (2004)
21. Preece, J., Rogers, Y., Sharp, H.: Design de Interação - Além da interação homem-computador. Bookman, Porto Alegre (2005)
22. Nielsen, J.: Topic: Heuristic Evaluation (1997), http://www.nngroup.com/topic/heuristic-evaluation/
23. Prates, R.O., Barbosa, S.D.J.: Avaliação de interfaces de usuário: conceitos e métodos. In: XXIII Congresso Nacional da Sociedade Brasileira de Computação, SBC, Rio de Janeiro, pp. 1–43 (2003)