Checking Web Accessibility with the Content Accessibility Checker (CAC)

Eduard Klein[1], Anton Bolfing[2], and Markus Riesch[2]

[1] E-Government-Institute, BUAS – Bern University of Applied Sciences, Bern, Switzerland
eduard.klein@bfh.ch
[2] «Access-for-All» Foundation, Zurich, Switzerland
{anton.bolfing,riesch}@access-for-all.ch

Abstract. The internet has become an indispensable tool for the access of information. However, most websites are not sufficiently accessible for people with disabilities. Accessibility problems originate from either the underlying CMS systems or from content authors disregarding fundamental accessibility requirements. With the Content Accessibility Checker CAC we give a tool at hand to specifically red-flag possible accessibility issues to authors. The checking criteria form a subset of the WCAG 2.0 standard and are published as a checklist for authors and publishers. CAC is available as a browser plugin and is published as open source on github. It is based on JavaScript and can be extended with specific checking rules. In checking mode it detects accessibility issues on a website, highlights it with an overlay in the web browser and gives hints and recommendations on improving web accessibility.

Keywords: Web Accessibility, E-Inclusion, Design for All.

1 Introduction

The World Wide Web is an essential tool for people when accessing information, using web applications, shopping on the internet, and participating in communication and social networks. For users with disabilities the user experience is very poor on most websites [1]. Even assistive technologies such as screen readers do not improve the situation because of poor semantic website structure or multimedia content without alternative representation. Conservative estimates suggest that 15-20% of internet users have sensory, motor or cognitive limitations. Taking into account demographic shifts with the first internet user generations entering more advanced ages, the proportion of people with various limitations surfing the internet must be expected to quickly rise in the near future.

As part of the EU funded TAO project (thirdageonline.eu), the German Wikipedia web site was assessed according to the WCAG accessibility guidelines (w3.org). The assessment was performed according to the three levels mentioned in the WAI (Web Accessibility Initiative) recommendations: expert based evaluation, conformity analysis, and user studies with people with disabilities. The resulting study [2] contains recommendations for improvement, which is published as part of a MediaWiki

K. Miesenberger et al. (Eds.): ICCHP 2014, Part I, LNCS 8547, pp. 109–112, 2014.

Accessibility Tracking Group[1] for the developer community, indicating the state of implementation.

The paper is structured as follows: after the problem description, a motivation for tool development is given, presenting requirements and evaluation of existing checking tools. Afterwards we explain the usage of CAC, its architecture and results achieved so far. The conclusion also suggests areas of possible further improvement.

2 Checking Web Accessibility

The Access for all foundation is campaigning for accessible ICT in Switzerland. It has established the most respected national website accessibility certification scheme and conducts expert reviews, WCAG 2.0 conformity tests and user tests with blind and multiply disabled people. From its studies[2], we could deduce that only about one percent of the most important websites to be WCAG2.0 accessible.

Typical accessibility issues are missing keyboard support and clean semantic site structure, where assistive technologies are further reliant on. Moreover, inappropriate contrasting color schemes and unclear heading structures in articles are a problem. Paragraphs and lists ought to be tagged as such; tables must be set up semantically correctly; table dimensions must be stated in the table headers. Alternative text for images must be provided. An extensive list and explanations are given in [3].

3 Comparison with Other Tools

We selected five popular tools (WAVE, Web Accessibility Inspector, Web Thing, Cynthia Says, Accessibility Toolbar) from an extensive set[3,] of tools for evaluation against the following functionalities:

- Checking during development: e.g. during editing or in preview mode
- Environment independence: analysis is HTML based (not CMS dependent)
- Category of messages: the tool shows messages on specific page elements, and gives recommendations which are valid for the whole website
- User experience: messages are shown as an overlay for better error localization
- Extendibility: user defined rules can be integrated

It turned out that none of these tools fulfilled all the requirements mentioned above; in particular none of them met the extendibility requirement. This led to the decision to implement the CAC tool.

---

[1] https://www.mediawiki.org/wiki/Groups/Proposals/Accessibility_Tracking

[2] http://access-for-all.ch/ch/publikationen.html

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

4 The CAC Content Accessibility Checker

CAC indicates accessibility issues which are caused by authors, and not by the CMS. The tool is based on the Accessibility Checklist for authors and publishers[4], comprising a subset of WCAG 2.0 criteria, which can be split in three categories:

- automatic test: based on the structure of the web page it can be decided if a criterion fails. Example: check hierarchical properness of heading levels.
- heuristic test: based on experience, the system makes a guess about failing of a criterion. E.g. for a download link (indicated through file suffix analysis), CAC recommends adding file size and format information.
- not testable (by a tool): it cannot be determined, if the description of an image sufficiently explains its contents. In this case, CAC gives a warning.

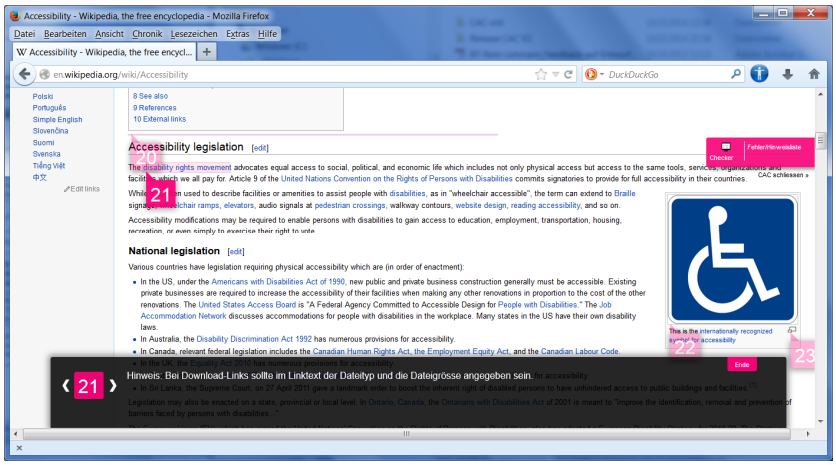

Fig. 1. CAC in action

CAC can be turned on/off as a browser plugin (see blue button in navigation bar in figure 1). If activated, the accessibility issues are presented as clickable overlay icons, at the bottom of the window the respective issues are explained in more detail. From the platform analysis in [3], it turned out that the realization in JavaScript promised the most benefits such as easy integration in web browsers and independence from a CMS. The checking process is solely based on HTML code.

The JavaScript code analyzes the structure of the web page based on its DOM by help of the popular jQuery library. It implements the checking rules, each of which consists of a pattern and an action. When a pattern is matched in a web page, i.e. the respective accessibility issue has occurred, an information item containing the location on the page and the error category is added to an intermediate storage in a

[4] http://access-for-all.ch/barrierefreiheit/barrierefreies-webdesign/checklist-2.html

de-fined format. This storage is realized as a set of DOM nodes which are rendered by the popular pageguide library.

CAC is open source (github.com/Access4all/ContentAccessibilityChecker) for further development. It comes in several distribution versions: as a Firefox plugin or as pure JavaScript code which can be integrated e.g. in a web page or a CMS.

5 Conclusion and Further Activities

CAC is to date the only tool providing accessibility error warnings to authors using accessible CMS. The approach to clearly differentiate between CMS and content regarding accessibility is novel. CAC helps authors identify possible and actual accessibility errors and helps raise awareness of accessibility issues. Moreover, it is helpful to prevent these issues by raising them on a strategic level for web projects.

Ongoing research and development on CAC are twofold. First, the evaluation of the tool with respect to quality of test results and implications on users/authors. It comprises the definition of evaluation criteria, their application to a set of representative web pages, and the implications of the evaluation.

Second, optimization and extension of functionality is ongoing at the time of writing, such as localization, the de-/selection of specific criteria, and optimizing the keyword list by which the CAC analysis is operated. Optimization of the heuristics for detecting possible accessibility issues is also planned. Concerning pattern based issue recognition, the use of a declarative pattern language has proven successful in many other tools and thus will also be integrated.

The information presented on the respective issues at the bottom of the page is planned to contain links with detailed information on the specific issues. Finally, the upcoming version of CAC will be available on the Mozilla Firefox Add-ons platform.

References

1. Kerkmann, F.: Web Accessibility. Informatik-Spektrum 36(5), 455–460 (2013)
2. Ritter, P., Riesch, M., Bolfing, A.: German Language Wikipedia Accessibility Test According to WCAG 2.0. Test Report. (2011), http://www.thirdageonline.eu/wp-content/uploads/2012/02/Accessibility_Test_Report_Wikipedia.pdf (accessed April 11, 2014)
3. Lehmann, R.: Entwicklung eines Accessibility-Assistenztools für Autoren von Online-Inhalten; Bachelor Thesis; BUAS-Bern University of Applied Sciences (May 2013)