UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Jaan Susi

Continuous Ranking of Estonian Public Sector Web Sites With Respect to WCAG 2.0 Guidelines

Bachelor's Thesis (9 ECTS)

Supervisor: Peep Küngas

Tartu 2016

Continuous Ranking of Estonian Public Sector Web Sites With Respect to WCAG 2.0 Guidelines

Abstract:

Accessibility of public sector Web sites has been recently recognized as one of the objectives of governments of EU member states and countries elsewhere. In order to measure in which extent the accessibility has been achieved, WCAG 2.0 guidelines have been adopted as a benchmark measure. Although conformance to the guidelines has been partially automated, there is still a lot of human effort and subjectivity involved in the evaluation process. Furthermore, due to human involvement, evaluation is mostly narrowed own to a limited set of Web pages of a domain under evaluation. This study aims to make another step toward evaluation automation by 1) reverse-engineering strategies of humans' evaluators and 2) analyzing whether higher number of evaluated Web pages will have positive effect to the final ranking. The experimental results show that human ranking is closer to semi-permissive and restrictive evaluation strategies. Furthermore, we show that higher number of evaluated pages will not have positive impact on the evalution precision wrt human rankings.

Keywords:

Web Archive, WCAG 2.0, Grading, Automated, Pa11y, Aggregation, Algorithms

CERCS: P170 Computer science, numerical analysis, systems, control

Pidev Eesti avaliku sektori veebisaitide hindamine WCAG 2.0 reeglite suhtes

Lühikokkuvõte:

Ligipääsetavus avaliku sektori veebilehtedele on hiljuti tunnistatud üheks eesmärgiks nii Euroopa liidu kui ka muude valitsuste poolt. Selleks, et ligipääsetavuse astet mõõta, on võetud mõõdupuuks WCAG 2.0 juhised, mille järgimise hindamine on pooleldi automatiseeritud. Kõikide kriteeriumite hindamine vajab siiski veel palju inimtööjõudu ja subjektiivsust. Inimressursi kokkuhoidmiseks piiratakse hindamine enamasti mõne leheküljeni iga domeeni kohta. Antud lõputöös püütakse parendada automatiseerimist: esiteks uuritakse inimhindajate strateegiat, teiseks analüüsitakse, kas suurema arvu lehtede hindamine ühes domeenis mõjutab lõplikku hinnangut. Eksperimentaalsed tulemused näitavad, et inimeste hindamine on lähemal pooleldi lubavale ja keelavale strateegiale. Kõrgema arvu lehtede hindamine ei mõju positiivselt hindamise täpsusele võrreldes inimeste hinnangutega.

Võtmesõnad:

Veebi arhiiv, WCAG 2.0, Hindamine, Automaatne, Pa11y, Agregeerimine, Algoritmid

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1 Introduction

The aim of this Thesis is to automate the grading of websites according to WCAG 2.0 rules and generate a system based on comparing automated and manual grading. This allows to give sites a grade based on only the automatically controlled points by looking at correlations between the controlled points and the final grade of the website.

Web Accessibility describes how easily the website can be used by people with different disabilities (bad eyesight, deafness, inability to use a mouse). The Web Accessibility Initiative (WAI) was founded by the World Wide Web Consortium (W3C) to promote the accessibility of the Web.

In Estonia, there is a web collaboration framework which aims to raise the quality of the public sector websites and their association regarding their user groups. Its objectives are:

- Raise the quality, actuality and make sites more in touch with the current times
- Create a structure standard for open sector websites and a mechanism for reusing them
- Change automatic retrieval and processing of data (for and from web sites) as easy and effective as possible

Included in the article is a chapter which states that a website must conform to the XHTML and CSS standards and conformance to these two standards must be checked regularly. In addition, to help involve people with disabilities into everyday life the sites must also follow WCAG 2.0 and WAI-ARIA standards.

To check the accessibility of Estonian public sector websites in accordance to the WCAG 2.0 guidelines, every 1-2 years, the Estonian Ministry of Economic Affairs and Communications orders a research on the subject of the public sector websites accessibility. This is conducted manually by experts, which means that every page is checked through manually for errors.

WAI-ARIA and WCAG 2.0 are standards created by the W3C. WAI-ARIA specifies how to increase the accessibility of dynamic content and components developed with Ajax, HTML, JavaScript and related technologies. WCAG 2.0 however, is a set of guidelines that specifies how to make content more accessible in general, primarily for people with disabilities but also for all user agents including for example, smartphones.

Evaluation of websites can be done in three different ways:

- experts systematically review the website,
- users fill out surveys about the website,
- automated tools are used to determine if internal attributes are correct.

These three can all be used together for maximum efficiency, but this thesis concentrates on the automated tools for grading.

For auditing, a tool called Pa11y, which makes use of the HTML_CodeSniffer library to assess pages offline, is used. Pa11y can be called upon from the command line which makes it easy to run in a Unix system. The created program uses that to get a programmatically parseable output which in this case is in json format. The data is read from the result of the audit, processed and uploaded to a database for every HTML page in every warc (Web Archive) that is a part of one the Estonian public sector domains. Pa11y is used three times, once for each WCAG 2.0 level. This is done on multiple threads simultaneously to speed up the process and use the server efficiently. The thread count is entered as

a command line argument for simplicity. 12 threads on 12 cores occupied about 70% of the given servers computing power so this amount was used to leave some room for other processes.

After collecting the data, a method of aggregation needed to be implemented. The important factor was to process obtained data so that it looks as similar to the output of human auditing as possible. For the human grading data sample, the year 2015 research of Estonian Public Sector websites was used [1]. The idea to use different approaches to aggregate data came from the work of Fuertes et al. [2].

Hera-FFX, a Mozilla Firefox extension created by Fuertes et al. used a permissive, restrictive and semi-permissive algorithms. These, however, were unfitting for this thesis since they did not only generate pass and fail results, but also verify, unknown, partial and not applicable. Those results are not relevant in this thesis and therefore this exact solution cannot be used. Instead, a more generalized solution is used instead of the semi-permissive algorithm which asserts a numerical value to different results. These are then analysed to generate a grade for the website.

2 Related work

W3C provides a definition for testable: “Either machine testable or reliably human testable”. Machine testable means that there is a known algorithm (implemented or not) that will determine, with complete reliability, whether the technique has been implemented or not. A technique is reliably human testable when it can be tested by human inspection and it is believed that at least 80% of knowledgeable human evaluators will agree on the conclusion [3].

In 2010, the challenges of evaluating sites in accordance to WCAG 2.0 were documented by Alonso et al. [4]. The first challenge comes from the fourth understanding requirement [5] and is that technologies should only use accessibility-supported ways. This says that every information or functionality that is provided in a way that is not accessibility supported is also provided in a way that is accessibility supported. A technology is considered accessibility supported when users’ assistive technologies and accessibility features in browsers and other user agents support it. The problem with this is that W3C does not specify the amount nor which assistive technologies a functionality should support to be accessible. From here, the first and main challenge: a definition of which technologies are considered to be “accessibility supported” in a given context is needed in order to get consistent evaluation results.

The second challenge deals with the actual testability of the success criteria. Many criteria have exact values to test against (e.g. 1.4.3 - colour contrast and 1.1.1 – non-text content) but many human testable criteria rely on a subjective opinion and therefore may not give accurate results (e.g. 1.3.3 – sensory characteristics). A research paper by Brajnik on the subject of the testability and validity of success criteria is discussed later in this thesis.

The third, openness of techniques and failures, comes from that W3C only documents techniques for accessibility for non-proprietary technologies [6] and hopes that vendors of other technologies will provide their own techniques on how to conform to WCAG 2.0. W3C encourages the submission of new techniques to their document and has established a process for updating it. This raises the issue: how to get a reliable result with an undocumented technique.

The last challenge: aggregation of partial results. In WCAG 1.0, the situation was simple: if one element fails to comply with a checkpoint, the entire page fails that checkpoint. In WCAG 2.0, there are potentially different ways of making some specific elements accessible. If none of the documented techniques pass for any given element, the web page could still be accessible due to an undocumented technique. It can also be difficult to aggregate values of the techniques because it is not clear if an ‘and’ or an ‘or’ operator should be used. The challenge here is that W3C has not documented how to combine the results of techniques, failures and situations to produce an aggregated result for each success criterion. This can lead to different evaluators using different aggregation methods and thus producing different results.

During the development of a tool named HERA-FFX [2], an aggregation method was thought out. Fuertes et al. came up with 3 different algorithms: restrictive, permissive, and semi-permissive. The program chooses the method depending on the type of elements considered. For instance, if the current item is a group of sufficient techniques linked with an OR and with no requirement for a minimum of positive results, the permissive algorithm is used.

The actual validity and testability of WCAG 2.0 was researched by Brajnik, Yesilada and Harper [7] by having experts and non-experts evaluate four web pages. The accessibility evaluation can be carried out in 5 different methods:

- Inspection methods – based on an evaluator inspecting a web page. Most popular is the conformance method where the evaluator uses a set of guidelines that focus on possible problems and has to decide whether the page complies with these or not.
- Automated testing – an evaluator using an automated tool
- Screening techniques - using a website by reducing artificially users abilities
- Subjective Assessment – evaluator hires autonomous users to send their opinions
- User testing – users are asked to navigate the page, their behaviour is observed by the evaluators.

They had 22 expert and 27 non-experts evaluate all 61 success criteria of WCAG 2.0 on 4 web pages using the conformance (falls under the inspection method) method. An ideal accessibility evaluation should always yield in accurate results of all the problems and no false positives. Their results show that with expert evaluators, half of the success criteria do not meet the 80% agreement threshold. They produced 26% of incorrect ratings, 20% false positives and missed 32% of actual problems. Agreement amongst non-experts was 10% lower, there were 32% of incorrect ratings, 42% of false positives and 49% of false negatives.

Brajnik also reviewed three fundamental processes that are ridden with potential traps which affect the reliability and validity of evaluations [8]. These are: selecting the pages for investigation, finding their problems, and measuring the accessibility levels. Knowing which traps are there and how to avoid or overcome them should be important for researchers and practitioners in accessibility. Brajnik and his students found that the error in accessibility, if the worst sampling method for the pages is used, is close to 20%. This means that if the wrong pages are chosen for audition, the result may be up to 20% false.

He claimed that:

- conformance to WCAG 2.0 and barrier walkthrough are subjective;
- even experts make a relatively large number of errors when determining
- conformance;
- manual metrics are the best choice but they are expensive; semi-automatic ones suffer from blindness with respect to false negatives;
- automatic means to measure accessibility are cheap, but not useful.

While the automatic tests are not useful in the general sense of grading a website completely, they can be used very effectively to root out periodic issues. This, in conjunction with an expert creates very efficient results. This thesis aims to find whether only a fully automated system can be created to assess pages well enough to mimic manual grading.

In Estonia, there have been three studies to check the accessibility of the public sector: 2010 by Axis Multimedia OÜ, 2013 and 2015 by Ernst & Young Baltic AS [9]. These studies were completed using both automatic and manual testing combined. This means that automatically testable success criteria are assessed by a program (e.g. does an image have an alt attribute and if it exists, does it have content; otherwise a screen reader skips the image which may leave out information for people who can't see the image), while other criteria, which require manual checking are checked over by an expert (e.g. is the content placed logically on the page or are the errors given precise and with enough information). For automatic testing, WAVE and Accessibility Valet were used. However,

this thesis focuses on the automation of such assessment and thus cannot make use of these tools. To clarify, these tools are for online use only and do not output result that is easily parseable with a program. Therefore, these tools are not used for this thesis.

On every checked domain at least the home page, search page and the contacts page were assessed and if the pages corresponded to the level A criteria, then search was expanded to include other pages. This was done to save time since most domains did not even pass level A on the first three pages so nothing else was audited.

To ensure a bigger diversity in result data some concessions were made because most of the domains did not reach level A. Compromised grades were given to domains whose mistakes were small and didn't cause greater issues with viewing the page and the segregating of content. These gave results as following:

- 2010 – level AA was achieved by 2 pages, level A by 9 pages. 1 with 1-2 mistakes, 13 with 3-4, 99 with 5-7, and 166 with more than 7 mistakes.
- 2013 – level AA was achieved by 3 pages, level A by 12 pages, 60 with 1-2 mistakes, 145 with 3-4 mistakes, 62 with 5-7, and 3 with more than 7 mistakes.
- 2015 – level AA was achieved by 7 pages, level A by 11 pages, 11 with 1-2 mistakes, 23 with 3-4 pages, 135 with 5-7 mistakes, and 93 with more than 7 mistakes.

In addition, every pages' HTML code was checked automatically with a validator created by W3C. Regularity in source code, intent of attributes and elements, and severity of errors were checked manually.

In Turkey, Akgül and Vatasever [10] conducted automatic tests on the 25 e-Government websites of Turkey. These tests were made with a wide variety of programs, 13 for accessibility testing and 4 for HTML and CSS validity. All domains that were assessed had accessibility issues already on the home page, only 1 did not have HTML errors as opposed to one having 525. They also found two main limitations when auditing without user testing. First, that automatic tools cannot replace user testing, because of the difficulties that understanding the interactions between web content and assistive technology present. Automatic tools generally verify the presence and value of a specific attribute or the validity of an element. However, whether that attribute is correct or carries the right meaning is left to the expert to decide. For example, there is a difference between an image that conveys content and an image that represents a certain elements background. It is mentioned that in some cases the image may actually not need an alternative text, whose presence an automatic tester checks. The second limitation was that they restricted testing to the home pages of domains, contrary to this thesis.

Akgül also emphasizes that these tests do not represent the real accessibility of a certain website but indeed act as an intermediate value and cannot give a real assessment of accessibility issues that disabled individuals may encounter. They only pinpoint to some major accessibility issues that need to be resolved. In addition, in this article, only open source tools were used. In conclusion, the authors recommend that the government should either adapt the existing guidelines or create new ones appropriate for them and set a policy for making the accessibility of government websites compulsory.

Adepoju and Shehu [11] studied the accessibility of Web sites for Nigerian Universities. They used aChecker, HERA and WAVE and found that 8% of university sites in Nigeria did not produce errors in accordance to WCAG1.0 with priority 1 (priority 1 violation will make it impossible for one or more group of people to access the website), every site

failed priorities 2 and 3 (similar to levels AA and AAA in WCAG 2.0). Every site produced errors when auditing with WCAG2.0. Most common errors were that of linked images missing, alternate text element missing, form label not present, document language not set, and empty links. They suggest that: awareness should be raised about accessibility, web developers should adhere to international principles, a regulatory body should mandate a compliance with WCAG, Human Computer Interaction as a course should be included in the computer science curriculum, at every stage of web development, periodic usability and accessibility evaluations should be carried out.

Vigo, Brown and Conway [12] investigated the effectiveness of 6 automated accessibility testing tools regarding the WCAG 2.0 conformance. They state that typically tools indicate a negative or positive result against a SC (success criteria) with no contextualized interpretation or its severity on accessibility. In the article, authors hypothesise that the existence of large amounts of domains which have low accessibility and still pretend to be accessible, by showing compliance logos, may indicate over-reliance on automated tests. In order to alleviate this lack of awareness they examine the role and reliability of these tools, and to what level an automatic assessment can be considered a representation of the web sites actual accessibility. They also discovered that the amount of human testers for most reliable and valid results is 3 in the case of experts and 14 in the case of non-experts.

Out of 26 SC violated, only 23-50% of SC were covered by automated tests in tools. A SC was considered to be covered if at least one true violation was reported by a given tool. TAW reported at least one true violation in 50% of the criteria, while the coverage of the remaining tools were smaller: 42% for Deque, 38% for SortSite, 35% for Total Validator, 31% for AChecker and 23% for AMP. 44-55% of tests can be automated. TAW, can only catch around 38% of violations. This means that in the best case scenario automated tests are able to adequately catch 4 out 10 accessibility violations. This score is drastically reduced by remaining tools until a 14%, which is the lower value exhibited for completeness. In general tools show a high level of correctness, which is higher than 93%. The low number of false positives contrasts with those found in similar studies where not only automated tests but also warnings were considered. Thus, we can say that tool developers do not take risks when it comes to automated tests and these are only reported under high levels of certainty. However, there are two exceptions, TotalValidator and TAW, which report 34% and 29% of incorrectness respectively. They conclude that if the right tool was used, about half of the errors would be caught but the result would worsen noticeably if a wrong tool was used.

For non-English web sites, a study was conducted for evaluating Arabic websites using automated WCAG 2.0 evaluation tools by Al-Khalifa et al. [13]. They used 5 tools to evaluate 9 different web sites. They found that some errors only emerged from evaluating an Arabic website. These errors were filed under SC 3.1. They also noticed that different tools returned very different results, sometimes varying 4 times in size. The authors concluded that accessibility checking is biased towards Latin-based websites and proposed that an automated tool that supports Arabic language and produces reports in Arabic should be developed.

Rattanavalee Maisak, in writing a doctors thesis [14], found out that for Search Engine Optimization (SEO), web developers are already required to meet few guidelines (1.1.1, 1.3.1 and 2.4.4). For example, developing a website following SEO standards means building the site with a clear, navigable structure in a way that the search engines can crawl effectively. In addition, there are many HTML5 elements which are machine-

readable for modern web browsers and assistive technology and so, developing websites with HTML5 can also lead to improved accessibility.

Brajnik tried to alleviate usability and accessibility issues by working out a method for updating and creating web sites and listing some already available programs to use [15]. He found that even simple changes to a web sites content can change the usability and/or accessibility of it. The most relevant reasons he stated are:

- Lack of resources, time, money or skilled persons, biggest problem being the lack of skills.
- Release cycles of web sites are too fast to be properly checked for issues. A website can be conceived, designed, implemented and released in a matter of hours.
- Detailed, accurate and complete specifications for the web pages are missing. When a web site is changed, unless the change is a significant redefinition of the site, often no analysis and specification steps will be carried out, at least not in full.
- Web browsing technologies are evolving fast, and even if the web site was designed to meet the standards at the time, it may be at some point obsolete and even hinder the usability of the website.
- Not all who create web sites, are educated in the accessibility and usability designs and implementations. Certain decisions which may seem trivial may lead to solutions that violate accessibility standards or usability guidelines.

He writes that a sites development and maintenance team includes at least the follo wing roles (could be carried out by fewer people): web architects, programmers, designers, content producers, accessibility and usability engineers, web masters and project managers. Achievement of accessibility and usability should not be seen as a goal to reach towards, but instead, as a process – like cleaning. One has to determine which rooms have to be cleaned at what level, how to determine whether they are clean, how frequently they have to be cleaned, and who should clean them with what.

Automatic tools cannot assess every aspect of the web page. In particular, anything that requires a subjective opinion (e.g. usage of natural and concise language) or that requires the assessment of relevance (e.g. ALT text of an image is equivalent to the image itself) is out of reach for programs. Nevertheless, these tools can find a number of issues that can later be checked by humans and can even avoid highlighting them when there is cause to think that the issue is not a problem. No automatic tool can be expected to thoroughly assess all of the websites accessibility problems. However tools can be used to better inform the decision maker and by providing functionalities that can help make decisions about giving grades easier.

Espadinha et al. conducted tests on 64 Portuguese university web sites 3 times during 3 years [16]. The work was conducted in three phases: identification of the websites, automatic checking of the homepages from the identified websites and search for correlations between accessibility claims of website designers and actual accessibility scores. For this, they used 3 different tools (Bobby, HERA, eXaminator) to assess the conformance in accordance to WCAG 1.0 guidelines. They found the presence of information regarding supporting services for disabled students in 12.5% of sites. The amount did not change during the 3 year testing period. The authors observed that the overall accessibility was not acceptable, even accounting in the significant improvement when comparing 2007 and 2008 data.

In Italy, Gambino, Pirrone and Di Giorgio [17] checked official web pages of the Italian Province to ascertain whether they were compliant to the 22 technical requirements de-

fined by the Stanca Act. Over the course of 3 months, they downloaded 976 web pages belonging to the websites of the Italian chief towns. These pages were submitted to Achecker for automatic analysis and in addition, HTML and CSS validity was checked. They found that in general, a less number of pages checked on a domain, does not imply a proportional decrease of the errors. One of the thigs that was suggested was that all domain addresses should be easy to remember. An ordinary rule consists of including the name of the brand or the company that makes it as a part of the URL. A similar use should be used for institutional websites. In some cases the format was www.comune.<name_of_city>.it where "comune" stands for town council while some of the web sites were even registered into different domains, like fi.it and net.it. They conclude that like many other countries, the Italian institutional websites are not accessible enough.

In Malaysia, Jati and Dominic [18] conducted tests on 90 websites from education, government and business sector, 30 from each. They used Tawdis to check for errors in accordance to WCAG 1.0 and found that 10 pages corresponded to level A, others failed in every aspect.

Hashemian in Finland [19] researched how many universities could achieve the level A conformance and how many violations could be found. Finnish government has published a Quality Criteria for Web Services (QCWS) for designing and assessing the quality of web services in the public administration which was last updated in 2008. This said that ideally a website should achieve level AA, but has to reach at least level A. In addition, it is recommended that the assessment of accessibility should be checked during implementation, production and maintenance of the site. For various reasons, the only tool that Hashemian was satisfied with was TAW Standalone. He checked 20 Finnish higher education institutes and found only 6 which conformed to the level A, none reached level AA. 11 sites failed level A with 1 error and the remaining 3 with 2 errors. He states that if the sites with 1-3 mistakes would fix these, 12 out of 20 websites could pass level AA.

	Scale of automation	Rulesets	Grade levels	Automation tools	Regularity	Web site coverage	Subjected domains count	Major reported errors
This thesis	Complete	WCAG 2.0	A, AA, AAA	Pa11y (HTML_CodeSniffer)	Possibly every day/week	Everything crawled	250	
Estonian public sector, E.	None	WCAG 2.0	A, AA, AAA		Every year	3- per domain	280	

& Y. B. AS [9]								
Arabic websites, Al-Khalifa et al. [13]	Complete	WCAG 2.0	A	Taw, AChecker, Worldspace FireEyse, WorldSpace, Web Accessibility Assessment Tool	Once	1 per domain	9	3.1
Benchmarking, Vigo et al. [12]	Complete	WCAG 2.0	A, AA	AChecker, SortSite, TotalValidator, TAW, Deque, AMP	Once	1 per domain	9	23-50% SC are covered by automated tests
Portuguese universities, Espadinha et al. [16]	Complete	WCAG 1.0		Bobby, HERA, eXaminator	3 times, yearly	1 per domain	64	
Italian institutional web pages, Gambino et al. [17]	Complete	Stanca Act	Inspired by WCAG level A	AChecker, CSS Validation Service, W3C Validator	Once	2-nd to 4th level depth	227	Requirements 15 and 16

Malaysia, Jati et al. [18]	Complete	WCAG 1.0	A, AA, AAA	Tawdis	Once	1 per domain	90	1.1, 3.4, 4.3, 5.5, 11.2
Finnish, Hashemian [19]	Complete	WCAG 1.0	A, AA	TAW3	Once	1 per domain	20	

Table 1 Comparison of different methods of grading websites in different studiesBackground

3 Background

3.1 Web content Accessibility Guidelines 2.0

The first major version of WCAG (Web Content Accessibility Guidelines) was published on 5th May 1999. It consists of 14 guidelines describing a general principle of accessibility. Each guideline covers some kind of theme for web accessibility and is associated with checkpoints to help apply them to specific features. In total WCAG 1.0 has 65 checkpoints which divide into 3 priorities, similar to the next major version, the WCAG 2.0, which has conformance levels A, AA and AAA.

WCAG 2.0 was published on 11th December 2008 as an upgrade from WCAG 1.0. It consists of four principles which house guidelines and these in turn, consist of success criteria. The WCAG 2.0 requirements are more precisely testable with automated tests and human evaluation. This allows it to be more simply used if specific requirements and conformance is necessary. To help conform to these guidelines, an extensive group of support materials was created for guidance including examples for easier understanding and ease of use.

Web Content Accessibility Guidelines 2.0 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content accessible to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these. The success criteria are written as testable statements that are not technology-specific [20]. Information on interpreting the success criteria and guidance on how to make a website conform to the rules is provided in several documents available at http://www.w3.org/WAI/intro/wcag.php.

WCAG 2.0 is divided into 4 principles:

Perceivable – Information and user interface components must be presentable to users in ways they can perceive.

Operable – User interface components and navigation must be operable.

Understandable – Information and the operation of user interface must be understandable.

Robust – Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

These principles are divided into guidelines. The 12 guidelines provide basic goals that web developers should work towards in order to raise accessibility to users with different disabilities. Guidelines themselves are not testable, being too generic, but provide overall objectives for implementing the techniques.

Under each guideline are testable success criteria which allow to check against certain specifications when they are necessary, such as in design specification or regulation. In order to satisfy different necessities and situations, three levels of guidelines are defined: A (lowest), AA, and AAA (highest).

“It is not recommended that Level AAA conformance be required as a general policy for entire sites because it is not possible to satisfy all Level AAA Success Criteria for some content.” [5]

For each of the guidelines and success criteria, a wide variety of techniques have been developed which fall into two categories:

- ones that are sufficient for meeting the success criteria
- others that are advisory.

Advisory techniques are meant to allow authors to better address the guidelines, but are not always testable by the success criteria. For further simplification, the most common areas of failure and how to avoid them are documented.

3.2 Pa11y

The technical limitation of the program created for this thesis is that it must be able to run on a Unix based server autonomously and parse ten thousands of warc files. This means that an offline tool should be used in order to not overuse the web connection since there are approximately 5 terabytes of data to process.

Pa11y makes use of the HTML_CodeSniffer library. This means that the auditing is still done by HTML_CodeSniffer functions, but the interface is new and more intuitive for command-line auditing. This thesis uses Pa11y instead of HTML_CodeSniffer for simplicity of the written program, since it does not matter which one to use in terms of results.

Why Pa11y? WAVE and Web Accessibility checker work online, the data which can be audited is a little short of 5 TB-s. Having said that, in about 50 thousand crawled warc files, there are over 5 million web sites which should all be audited. That means over 5 million requests to the server. HERA is for WCAG 1.0. SortSite and TotalValidator, which have graphical user interfaces are for testing individual pages/sites so that a human tester knows what to not check or what to check more carefully. These programs are not meant for mass auditing. Although a command line tool version of SortSite exists, it is meant only for a web developer to integrate it into used build environment. T.A.W. only has a graphical user interface and desktop version supports only WCAG1.0. Deque has a JavaScript library, which can be integrated to a testing framework which is out of the scope of this thesis. AMP is commercial and only has a web-based platform.

4 Workflow

The programs workflow was created so that if necessary, additional parsing programs or plugins could be easily attached. The page is read from the warc, then graded by appointed programs, and uploaded to the database. Every step uses a different class for clarity and independence so one part of the program could be updated without breaking other classes workflow. The source code for the program is included with this thesis.

4.1 Audited Criteria

In the following table are all the success criteria that HTML_CodeSniffer claims to grade. The last column describes what level the errors found describe under this success criteria. Notice that there are two success criteria which do not have levels graded. This is because on the pages parsed, these errors were not found and thus, could not be confirmed if these errors are actually found by the program. The list is taken from the HTML_CodeSniffer website and contains all the success criteria under which, at least a warning could be found.

Although HTML_CodeSniffer allows for 4 different answers: error, warning, notice, pass; for this thesis, notices are irrelevant since they are meant for just directing attention of a human grader. The difference between a warning and a notice is given in the program standards web page [21]:

"Warnings are items that HTML_CodeSniffer are able to detect as a potential problem, but require manual inspection to determine whether it is a failure."

"Notices are items that HTML_CodeSniffer cannot automatically detect, but should be manually inspected to ensure compliance with the standard."

Success criteria	Description	W3C documented techniques failure	Levels graded
1.1.1	All non-text content that is presented to the user has a text alternative that serves the equivalent purpose.	Failure	A, AA, AAA
1.3.1	Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.	Failure	A, AA, AAA
1.4.3	The visual presentation of text and images of text has a contrast ratio of at least 4.5:1.	Failure	AA, AAA
1.4.6	The visual presentation of text and images of text has a contrast ratio of at least 7:1.	Failure	AAA
2.1.1	All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires	Failure	A, AA, AAA

	input that depends on the path of the user's movement and not just the endpoints.		
2.1.2	If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away.	Failure	A, AA, AAA
2.2.1	For each time limit that is set by the content, at least one of the following is true: turn off, adjust, extend, exception.	Failure	A, AA
2.2.2	For moving, blinking, scrolling, or auto-updating information, all of the following are true: there is a mechanism for the user to pause, stop, or hide it.	Failure	Could not be determined
2.4.1	A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.		A, AA, AAA
2.4.2	Web pages have titles that describe topic or purpose.	Failure	A, AA
2.4.8	Information about the user's location within a set of Web pages is available.		AAA
3.1.1	The default human language of each Web page can be programmatically determined.		A, AA, AAA
3.1.2	The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.		AA, AAA
3.1.6	A mechanism is available for identifying specific pronunciation of words where meaning of the words, in context, is ambiguous without knowing the pronunciation.		Could not be determined
3.2.2	Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behaviour before using the component.	Failure	A, AA, AAA
3.2.5	Changes of context are initiated only by user	Failure	AAA

	request or a mechanism is available to turn off such changes.		
4.1.1	In content implemented using mark-up languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique.	Failure	A, AA, AAA
4.1.2	For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined.	Failure	A, AA, AAA

Table 2Success criteria graded by HTML_CodeSniffer

4.2 Database model

Because the data generated in this thesis is very straightforward, the model for this database was created with simplicity and readability in mind. All values were stored in a single table so they would be easily readable with a client and at the same time easily found with the aggregation program. The SQL for the database has been uploaded with the other programs for reference.

4.3 Data aggregation

Data that this thesis compares results against comes from the 2015 evaluation of Estonian public sector websites [9]. The human conducted study found that:

- 1 website reached level AA
- 6 websites reached level AA with one or two minor mistakes
- 1 website reached level A
- 10 websites reached level A with one or two minor mistakes
- The remaining 262 websites did not pass the audit

This system of aggregating single success criteria results into a grade is restrictive but with some concessions. This means that once a single page fails a success criteria in a domain, the whole domain is considered failed in case of this criteria. Since one or two small mistakes were allowed, this system cannot be called fully restrictive. The reason for this was that when using a fully restrictive grading system, only 2 websites would have passed at all. In order to bring a little more diversity into the results, the concessions were introduced.

This thesis studies the result relation to three different grading methods.

First, permissive aggregation. This method of grading is very forgiving, once a single page passes a certain success criteria, that criteria is considered passed on the whole domain. This is generally used when auditing a module based website like blogs or news sites which have different content but same methods and outlook on different pages. Then, a generalization about the whole domain can be made since the whole site uses same techniques and technologies.

Second, restrictive aggregation. This method is very unforgiving. Once a success criteria fails, the whole domain fails at that point and is unable to receive the level of that success criteria. For example, if one page out of 10 graded fails at success criteria on level AA the

whole domain can only reach the level A, even when all the other success criteria at every level passed.

Last, semi-permissive aggregation. This method works in-between the other two. Error, warning and pass are given a numerical value which is used to calculate the average on a website. Then, based on a limit which the score needs to surpass, this number determines whether the success criteria passes or fails within this domain. The collective grade for a site is determined by the same system as the original study to which this thesis tries to find a supplement. Restrictive, while allowing one or two mistakes. A third failed success criteria in the same level already restricts the grade.

5 Results

Domains that were found in the timeframe of parsed warcs varied greatly in numbers. 5000 warc files that contained the pages from 2015-05-01 to 2015-05-17 were parsed. In these, amongst other domains that were not graded, were the pages of 43 domains. The page count of domains varied greatly, from 1 page of pihtlavv.ee to 22702 pages of sangaste.ee. This introduced a disparity into the data, when concerning the domains which had less than 15 pages as seen in the chart 1.

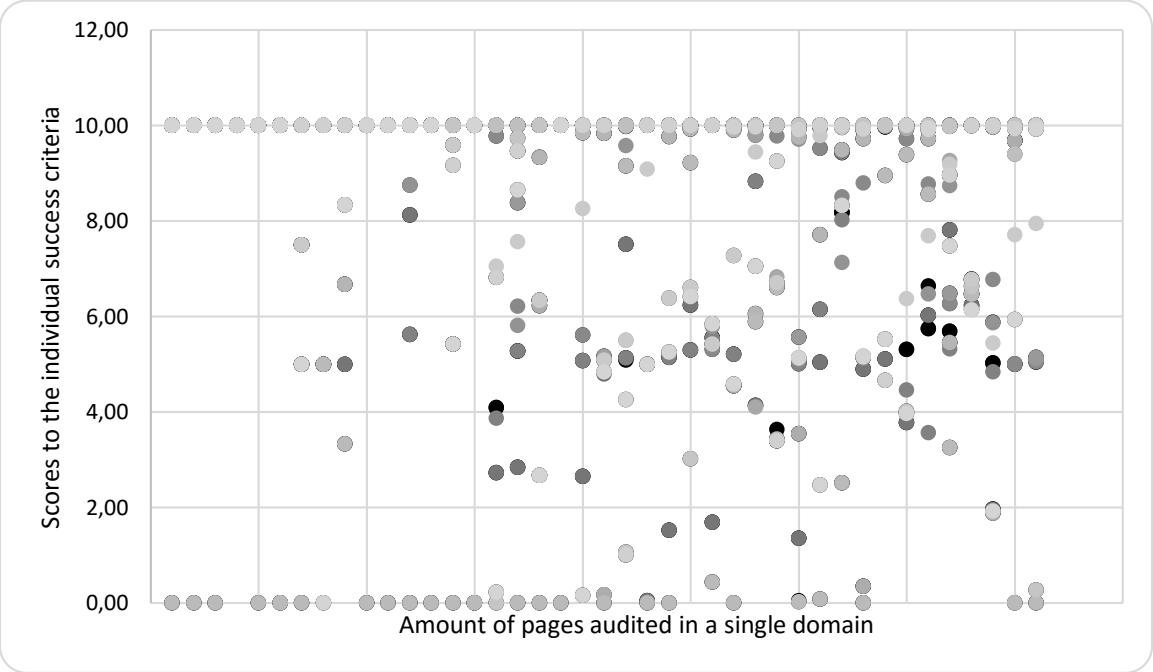

Chart 1Semi-permissive algorithm, results of domain scores on individual success criteria in relation to page count

This introduced an interesting phenomena, the domains with up to 15 pages had only scores of either 0, 5 or 10 (pass, warning or fail) which meant that, for example, if a page had a success criteria error, all the other pages on the domain had the same result too. Out of the 15 sites that had under 15 pages, using the semi-permissive scoring values, 11 got the level AAA, while the rest 4 failed. To compare, of the 28 sites that had over 15 pages, 3 reached the level AAA, 1 reached level A and the rest failed.

Domain	Page count	Permissive	Restrictive	Semi-Permissive	Manual
konguta.ee	1	AAA	AAA	AAA	-
pihtlavv.ee	1	AAA	AAA	AAA	-
vonnu.ee	1	AAA	AAA	AAA	-
aserivv.ee	2	AAA	AAA	AAA	-
illuka.ee	2	AAA	AAA	-	-
kuusalu.ee	2	A	-	-	-
sonda.ee	2	AAA	-	AAA	-
vald.koo.ee	2	AAA	AAA	AAA	-
vandra.ee	3	AAA	-	-	-
kohtla-jarve.ee	4	AAA	AAA	AAA	-
eestipank.ee	7	AAA	AAA	AAA	-
tostamaa.ee	8	AAA	-	AAA	-
tallinn.ee	11	AAA	AAA	AAA	-
pria.ee	12	AAA	-	-	-
politsei.ee	15	AAA	AAA	AAA	-
muinas.ee	22	-	-	-	-
moisakyla.ee	45	AAA	-	-	-
kadriorg.ee	50	AAA	AAA	AAA	-
paldiski.ee	66	AAA	-	-	-
muhu.ee	120	AAA	-	-	-
luunja.ee	189	AAA	-	-	-
meeksi.ee	202	-	-	-	-
helme.ee	253	AAA	-	-	-
otepaa.ee	268	AAA	-	-	-
rakke.ee	278	AAA	-	-	-
vandravald.ee	288	AAA	-	-	-
veriora.ee	302	AAA	-	-	-
vormsi.ee	359	AAA	-	-	-
voru.ee	452	AAA	-	-	-
ruhnu.ee	481	AAA	-	-	-
rannu.ee	486	AAA	-	AAA	-
iisaku.ee	544	AAA	-	-	-
kihnu.ee	582	AAA	-	-	-
ambla.ee	648	AAA	-	-	-
varstu.ee	937	AAA	-	A	-
vaivara.ee	1303	AAA	-	-	-
raikkyla.ee	1366	AAA	-	-	-
ravimiamet.ee	2507	AAA	-	-	-
puka.ee	5447	AAA	-	-	-
uus.saue.ee	5709	AAA	-	-	-
vm.ee	16244	AAA	-	AAA	AA*
sangaste.ee	22702	AAA	-	-	-

Table 3Results for different algorithms in relation to the page count when grading all crawled pages.

Domain	Page count	Permissive	Restrictive	Semi-Permissive	Manual
ambla.ee	3	-	-	-	-
eestipank.ee	1	AAA	AAA	AAA	-
helme.ee	2	-	-	-	-
iisaku.ee	2	AAA	-	-	-
illuka.ee	1	AAA	AAA	AAA	-
kihnu.ee	1	-	-	-	-
kuusalu.ee	1	-	-	-	-
luunja.ee	2	-	-	-	-
meeksi.ee	2	-	-	-	-
moisakyla.ee	1	-	-	-	-
otepaa.ee	2	-	-	-	-
puka.ee	1	-	-	-	-
raikkyla.ee	2	A	-	-	-
rakke.ee	2	-	-	-	-
rannu.ee	2	AAA	AAA	AAA	-
ravimiamet.ee	2	-	-	-	-
ruhnu.ee	2	-	-	-	-
tallinn.ee	1	AAA	AAA	AAA	-
uus.saue.ee	1	-	-	-	-
vaivara.ee	2	-	-	-	-
vald.koo.ee	1	AAA	AAA	AAA	-
vandravald.ee	1	-	-	-	-
varstu.ee	2	A	-	-	-
veriora.ee	2	A	-	-	-
vm.ee	3	AAA	AAA	AAA	AA*
vormsi.ee	2	-	-	-	-
voru.ee	1	-	-	-	-

Table 4Results with different algorithms in relation to the page count when grading a small portion of pages

5.1 Permissive

Permissive algorithm, where, if one page passes a criteria, all do, gave results as expected. 39 sites out of 42 reached level AAA. The interesting websites to note here, are the two that got no grade, despite having more than 15 pages audited. This means that in every page of the domain meeksi.ee (there were 202 pages graded), there were errors present in every page consistently.

This result is very different from the manually graded study, even when not considering the grades of specific websites. Here 93% of the sites received the grade AAA whereas in the manual study, none of the pages reached this level.

To compare with different results, a more similar page count to the human audit was taken. For most of the sites in the manual study, only 3 pages on each site were used. 27 pages that had home pages crawled for this thesis were taken out of the 42 used in this study. For these pages, if possible, the search and contact page were taken along with the home page and then graded. As thought, because of the low count of pages in the second data set, the permissive algorithm gave much lower grades. In contrast to the other permissive algorithm aggregation, this data set gave a level AAA to only 26% of websites and level A

to 11% which, when comparing to the manual study, bears a much closer resemblance to the manual audit but still statistically misses by 30%.

5.2 Restrictive

Restrictive algorithm where, if one success criteria fails on one page, the whole site is considered to have failed that success criteria. This algorithm generated a more similar output to the original data than permissive, especially if sites that had under 15 pages were ignored. Out of the 11 pages that passed, and not only passed, but also got a level AAA, 10 had a page count of under 15. If disregarding small page count websites, the result is that 96% of sites failed to meet even level A success criteria. This is much closer to the 94% failure that was received by the manual grading. However, the domain results do not match while comparing the levels achieved. Kadriorg.ee which received the grade AAA, failed in the human audit and vm.ee which did not pass this study, received a level AA in the human audit. This suggests that while being statistically almost correct (2% error), the results still do not correlate with the original study. Sadly, no other websites that achieved a level in the original study were represented with a grade in this set.

Surprisingly, results for the restrictive algorithm when using a small data set produced the exact same results as the semi-restrictive algorithm on a small data set. Unsurprisingly, using a small data set gives better grades with a restrictive algorithm compared to the big data set. This is caused by the nature of the algorithm, the less pages where something can go wrong, the better the grade. However, this differentiates more from the human study, with 78% of the websites failing to get a grade, than the data set with all the pages.

5.3 Semi-Permissive

The semi-permissive algorithm is the one with the most playing ground. First, the values of different results can be altered (i.e. warning does not give 5 out of 10 points as in this study, but only 3). Second, success criteria could be given different weight ratios (e.g. A1.1.1 error is worth less than A2.1.1 when calculating the final grade). Third, the limit on which a certain success criteria is considered passed when reading through the averages of domain (in this study 8 points out of 10 was considered a pass). This means that when the sites average pass/fail ratio has been calculated for each success criteria, it is compared to a predetermined number (e.g. 8). When the score does not reach the limit, the criteria is considered failed. This could also be implemented individually, i.e. each success criteria has its own limit. Lastly, with every algorithm, the allowed mistakes count can be changed for different levels. Currently it is set to 2 at every level, but can be easily changed. As seen in table 3, the sites with less than 15 pages still give false positives for most of the sites in this case. To counter this, the small page count can be taken into account to then allow for less mistakes which could give a more accurate depiction of the real grade. When looking at statistics according to sites with the page count over 15, this algorithm considered only 15% of the domains passed. This, while being more accurate than the permissive algorithm, still did not reach the statistical accuracy of the restrictive algorithm. However, this algorithm, unlike the restrictive one, did not produce a false negative when auditing all the pages of the site vm.ee.

As mentioned, the results when considering the small data set, coincide with the results of the restrictive algorithm exactly. This suggests that when using a small data set, these two algorithms act the same way, while using current values in the semi-permissive algorithm. When changing variables in the semi-permissive algorithm, the results may change. Compared to the 15% of sites that were given a grade with the big data set, the small set

achieved a 22% pass rate. If comparing the audits from both big and small data sets with the manual study, the algorithm which used all crawled pages got a more accurate statistical grade ratio than the one which used only a few pages of every site.

6 Conclusions

This study was made to better understand if an automated system can grade a website while giving results which are relevant when comparing to a human evaluators study. A program was created to automatically grade previously parsed warc files which were uploaded to a database. A different program was created to analyse the data and try to provide as close approximations to human evaluators as possible. For this, three different techniques were used which yielded different results: permissive, restrictive and semi-permissive. In order to check whether the amount of pages graded affected the grades, a secondary data set was introduced. This included 27 of the original 42 sites which were graded for this thesis. The entries of these 27 pages were manually inspected to find the home, contacts and search pages. These pages were picked for their use in the manual inspection. In the manual study of Estonian public sector websites, at least these three pages were checked on every site.

With the larger data set, the most inaccurate algorithm, unsurprisingly, turned out to be the permissive algorithm which is because of the nature of this study. One domain reached 22 thousand files audited which makes it highly unlikely that certain success criteria failed on all the pages. 39 domains out of 42 reached the level AAA, 1 reached level A and two failed. Surprisingly, one of the domains that failed had over 200 pages graded but upon visiting the site, it turned out that the website was built so that the content changed in the middle of the page but the menu and sidebar remained constant. These probably had an error in them which registered on every crawled page. Looking at the scores with the semi-permissive algorithm for this site confirmed that indeed one success criteria received a pure 5.0 which is very unlikely to happen in the balance of passes (score 10.0) and failures (score 0.0). More likely, every page had a warning flag which caused it. Upon searching in the database for it, this assumption was confirmed. One success criteria on the level A was also consistently failing, along with many higher level success criterias. On the brighter side, this modular build shows that if all the problems in that sites menu and sidebar could be fixed, the site would probably get a very high grade due to the nature of the pages being all created from a single blank. Only the errors in the blank would have to be fixed in order to fix all the pages (assuming that the pages are dynamically created, not statically).

Second was restrictive algorithm. Surprisingly, it had only a 4% difference from the human audit. This happened due to the very low grades given to the sites of Estonian public sector websites. When auditing over a thousand different pages in a single site, it is not realistic to assume that all of them have perfect scores and mistakes in a maximum of 2 success criteria per level which need to be consistent in the whole domain. If the human audited data would have turned up more positive, e.g. 50% of the pages reached at least level A, the mistake margin from the human audit would have probably been the same as with the permissive algorithm, just to the other side, although this needs to be checked. When the permissive algorithm passed almost every site, the restrictive one fails almost every site, especially when one site provides over hundreds of pages. As discussed in the previous paragraph, it is possible for a site to achieve a level even when big amounts of pages are graded, but that is realistically done only with a modular build, where on different pages content changes but everything beside the content originates from one place which has been built with accessibility in mind.

Last, the semi-permissive algorithm. This was deemed the most accurate due to the high flexibility of its grading and aggregation of data and due to the fact that this algorithm was

the one that, statistically speaking, deviated by 8.5% from the ratio of failed pages in the manual study. In addition, this algorithm also graded one site with a level AAA which achieved the level AA in the manual audit. Compared to restrictive, this showed that there can be some relation between manual and automatic grading, especially if the automatically parsed sites had a wider range of pages to grade.

When grading the smaller data set, the results of semi-permissive and restrictive algorithms matched exactly. They both gave the level AAA to 22% of the domains, while the permissive algorithm gave 26% of sites an AAA and 11% a level A. This showed that the high amount of passed domains when using the permissive algorithm was indeed dependent of the large amount of pages graded. The same effect, although less pronounced, happened with the restrictive algorithm, which gave a passing grade to 18% more domains. The smallest change in the statistical accuracy was with the semi-permissive algorithm. When grading less pages per site, the percentage of domains that were given a grade raised by 7%.

In table 5, in addition to the percentage of domain grades, precision, recall and their f-score has been included. Precision shows the ratio of correct non-failing results to all the passing grades given. Recall shows the amount of correct non-failing grades given in relation to the manual audit grades. F-score is the harmonic mean of the two.

Strategy	Small data set	Big data set	Manual audit
Permissive	26% AAA, 11% AA Precision 10% Recall 100% F-score 0,18	93% AAA Precision 2,5% Recall 100% F-score 0,05	2.5% AA 4% A
Restrictive	22% AAA Precision 17% Recall 100% F-score 0,29	3.7% AAA Precision 0% Recall 0% F-score 0	
Semi-permissive	22% AAA Precision 17% Recall 100% F-score 0,29	11% AAA 4% A Precision 6,7% Recall 100% F-score 0,13	

Table 5Results of different data sets vs human audit

Further study should be made with a complete set of pages where every human graded site also has at least 30 pages for automatic grading. These do not have to be manually looked through, but the ones that are manually graded, must be in the set. This would ensure that the pages graded by experts are all also audited automatically and the data set, at least to a part, is the same. Currently, the crawled warc files contained at least 10 times more sites than the ones this study needed and during the course of 17 days of crawling, only 43

websites were crawled out of the 280 in the human study. It is recommended that the next time a human study is carried out, the experts that grade websites at least save the pages they graded. This way, this program could be run on a small, but precise collection of files which would yield more accurate results. Right now, this study was conducted using the pages that were crawled on the same time that this study was conducted, the pages could have been altered or some sites could have been graded on the pages that were not even looked at in the human audit. This introduces a possibility that some of the results found in this study would not even correlate with the manual study when manually graded.

In addition, a learning algorithm should be included into the created program which would make use of the semi-permissive algorithms variables and find the most accurate scores to compare sites and pages to.

7 References

[1] Ernst & Young Baltic AS, "Avaliku sektori veebilehtede vastavus WCAG 2.0 nõuetele 2015. aastal," 2015. [Online]. Available: https://www.mkm.ee/sites/default/files/wcag_uuringuandmed_2015.xlsx. [Accessed 05 05 2016].

[2] J. L. Fuertes, E. Gutiérrez and L. Martínez, "Developing Hera," 28 03 2011. [Online]. Available: http://dx.doi.org/10.1145/1969289.1969294. [Accessed 26 04 2016].

[3] M. Cooper, "Requirements for WCAG 2.0 Checklists and Techniques," 07 02 2003. [Online]. Available: https://www.w3.org/TR/wcag2-tech-req/. [Accessed 13 04 2016].

[4] F. Alonso, J. L. Fuertes, Á. L. González and L. Martínez, "Evaluating Conformance to WCAG 2.0: Open Challenges," 14 07 2010. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14097-6_67. [Accessed 24 04 2016].

[5] "Understanding Conformance | Understanding WCAG 2.0," [Online]. Available: https://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html.

[6] M. Cooper, A. Kirkpatrick and J. O Connor, "Techniques for WCAG 2.0," 17 03 2016. [Online]. Available: https://www.w3.org/TR/WCAG20-TECHS/. [Accessed 25 04 2016].

[7] G. Brajnik, Y. Yesilada and S. Harper, "Testability and validity of WCAG 2.0: the expertise effect," 2010. [Online]. Available: http://dx.doi.org/10.1145/1878803.1878813.

[8] G. Brajnik, "The Troubled Path of Accessibility Engineering: an Overview of Traps to Avoid and Hurdles to Overcome," 2011. [Online]. [Accessed 19 04 2016].

[9] E. &. Y. B. AS, "Avaliku sektori veebilehtede vastavus WCAG 2.0 nõuetele 2015. aastal," 2015. [Online]. Available: https://www.mkm.ee/sites/default/files/wcag_aruanne_2015.pdf.

[10] Y. Akgül and K. Vatansever, "Web Accessibility Evaluation of Government Websites for People with Disabilities in Turkey," 2016. [Online]. Available: http://www.joams.com/uploadfile/2015/0407/20150407052826694.pdf.

[11] S. A. Adepoju and I. S. Shehu, "Usability evaluation of academic websites using automated tools," 2014. [Online]. Available: http://dx.doi.org/10.1109/INNOVATIONS.2011.5893835.

[12] M. Vigo, J. Brown and V. Conway, "Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests," 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2461124.

[13] H. S. Al-Khalifa, M. Al-Kanhal, H. Al-Nafisah and N. Al-Soukaih, "A pilot study for evaluating Arabic websites using automated WCAG 2.0 evaluation tools," 2011. [Online]. Available: http://dx.doi.org/10.1109/INNOVATIONS.2011.5893835.

[14] R. Maisak, "Accessibility of Thai university websites: Awareness, barriers and drivers for accessible practice," 2015. [Online]. Available: http://ro.ecu.edu.au/theses/1715/.

[15] G. Brajnik, "Using Automatic Tools in Accessibility and Usability Assurance Processes," [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-30111-0_18.

[16] C. Espadinha, L. M. Pereira, F. Moreira da Silva and J. B. Lopes, “Accessibility of Portuguese Public Universities' sites,” 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.3109/09638288.2010.498554.

[17] O. Gambino, R. Pirrone and F. Di Giorgio, “Accessibility of the Italian institutional web pages: a survey on the compliance of the Italian public administration web pages to the Stanca Act and its 22 technical requirements for web accessibility,” 2014. [Online]. Available: http://link.springer.com/article/10.1007/s10209-014-0381-0.

[18] H. Jati and D. D. Dominic, “Website accessibility performance evaluation in Malaysia,” [Online]. Available: http://dx.doi.org/10.1109/ITSIM.2008.4631587. [Accessed 19 04 2016].

[19] B. J. Hashemian, “Analyzing web accessibility in Finnish higher education,” 2011. [Online]. Available: http://dx.doi.org/10.1145/2047473.2047475.

[20] B. Caldwell, M. Cooper, L. G. Reid and G. Vanderheiden, “Web Content Accessibility Guidelines (WCAG) 2.0,” 11 12 2008. [Online]. Available: https://www.w3.org/TR/WCAG20/.

[21] Squiz, “WCAG 2.0 Standard: Summary - HTML_CodeSniffer,” [Online]. Available: http://squizlabs.github.io/HTML_CodeSniffer/Standards/WCAG2/. [Accessed 27 04 2016].

[22] M. Hassanzadeh and F. Navidi, “Web site accessibility evaluation methods in action: A comparative approach for ministerial web sites in Iran,” 2010. [Online]. Available: http://dx.doi.org/10.1108/02640471011093499.

[23] A. Khan, H. Idrees and K. Mudassir, “Library Web sites for people with disability: accessibility evaluation of library websites in Pakistan,” 2015. [Online]. Available: http://dx.doi.org/10.1108/LHTN-01-2015-0010.

[24] P. Koutsabasis, E. Vlachogiannis and J. S. Darzenta, “Beyond Specifications: Towards a Practical Methodology for Evaluating Web Accessibility,” 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=2019120.

License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Jaan Susi** (date of birth: 26.01.1994),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
    1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
    1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Continuous Ranking of Estonian Public Sector Web Sites With Respect to WCAG 2.0 Guidelines,

supervised by Peep Küngas, Ph.D.,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **13.05.2016**