A Lightweight Methodology to Improve Web Accessibility

Mardé Greeff
Meraka Institute of the CSIR
P.O. Box 395
0001 Pretoria, South Africa
+27 12 841 3523
mgreeff@csir.co.za

Paula Kotzé
Meraka Institute of the CSIR and Institute for ICT Advancement, Nelson Mandela Metropolitan University
P.O. Box 395
0001 Pretoria, South Africa
+27 12 841 4791
paula.kotze@meraka.org.za

ABSTRACT
This paper introduces a methodology to improve the accessibility of websites with the use of free so-called automatic tools. The methodology has three iterative phases, namely assessing a website against accessibility guidelines, user testing and creating in-house 'guidelines' to prevent similar mistakes in future versions of the system. Aspects of accessibility addressed include the use of colour, accessibility guidelines and priorities, readability or comprehensibility, and screen reader simulators. We recommend free tools for each of these accessibility aspects and discuss the process that should be followed when evaluating a website.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: *Evaluation/methodology, Screen design, User-centered design;* H.5.3 [**Group and Organization Interfaces**]: *Web-based interaction*.

General Terms
Design, Human Factors.

Keywords
Automatic accessibility evaluation, disabilities, usability evaluation.

1. INTRODUCTION
Worldwide, legislation is being introduced requiring products, such as interactive computer-based systems, to adhere to a set of criteria with regards to accessibility for persons with disabilities. Four sets of 'guidelines' or legislation are currently prominent in accessibility research and used in practise to determine adherence, namely: 1) The World Wide Web Consortium (W3C) Web Content Accessibility Guidelines (WCAG) [28]; 2) Section 508 of the US Rehabilitation Act [26]; 3) The Barrierefreie Informationstechnik Verordnung (Barrier free Information Technology Ordinance) or BITV [3]; and 4) The Fujitsu Web Accessibility Guidelines [10]. Several initiatives are currently underway aiming to assist developers in ensuring that their systems/products meet these requirements. This includes a large number of so-called automatic tools that can be used to evaluate interactive computer-based systems, especially websites. Web accessibility, according to W3C, means that people with disabilities and older people 'can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web' [28]. All disabilities that affect access and use of the Web are incorporated in this definition, including visual, auditory, physical, speech, cognitive, and neurological disabilities. Developers of interactive systems, especially websites, are frequently not specialists in accessibility and usability techniques. We argue that an appropriate accessibility evaluation methodology, accompanied by supporting tools, guiding them through the development process may assist in developing websites with improved accessibility.

This paper describes a lightweight methodology making use of free accessibility tools that can be used by non-accessibility experts to ensure improved accessibility of an interactive system. Such a methodology will be particularly useful in a developing country context, where many organizations and user communities face a range of challenges, such as limited funding to purchase commercial accessibility evaluation tools and developers with limited skills, especially with regards to accessibility and usability. Project teams often have limited access to users that meet the end-user profile and who can take time out to do the evaluation. Users regularly have inadequate means to attend the evaluation. The use of varied assistive devices may cause further problems during the evaluation, especially due to the fact that the mastery of a particular device does not mean an automatic skills transfer to another or newer device, as well as hardware incompatibility [14].

We used an information portal specifically developed for people with disabilities as case study to develop the methodology and to test various free accessibility tools to determine what they can offer the developer. User testing involving users with a variety of disabilities was also performed on the portal and these results were compared with that of the automatic tools to inform the refinement of the methodology.

Section 2 presents background information about accessibility guidelines and legislation, as well as other related work. Section 3 discusses the proposed methodology, while section 4 highlights issues from the case study where the methodology was used on the information portal. Finally, section 5 concludes.

2. BACKGROUND
There are four sets of 'guidelines' that are prominent in accessibility research worldwide and which are used in practice to determine adherence. As introduction we briefly present some of

these guidelines, and initiatives to introduce legislation to comply with accessibility guidelines. We also refer to some related work by other researchers working in the same arena.

2.1 Accessibility Guidelines and Legislation

2.1.1 W3C

The World Wide Web Consortium (W3C) is an international consortium with the aim to develop Web standards [28]. It pursues its mission through the creation of guidelines that will lead to Web standards. W3C's Web Accessibility Initiative (WAI) focuses on enabling people with disabilities to more readily create and interact with web content. WAI promotes the implementation of existing accessibility guidelines in advanced authoring tools and improved evaluation tools that will enable automated support for development and repair of accessible web sites. Within WAI the Web Content Accessibility Guidelines (WCAG) documents were developed (of which WCAG 1.0 is still widely used, whilst WCAG 2.0 is currently being worked on). The documents explain how to make web content accessible to people with disabilities. The guidelines address two general subjects, namely ensuring graceful transformation and making content comprehensible and navigable. Each guideline has a number, guideline statement, rationale, and checkpoint definitions explaining how the guideline applies in typical content development scenarios. Each checkpoint has a 'Priority Level' based on the checkpoint's impact on accessibility: Priority 1 which must be satisfied, i.e. a basic requirement for some groups to be able to use the web; Priority 2 which should be satisfied, i.e. one or more groups will find it difficult to access information in the document; and Priority 3 which a web content developer may address, i.e. one or more groups will find it somewhat difficult to access information in the document. The checkpoints are used to specify levels of conformance: Level A where all Priority 1 checkpoints are satisfied; Level AA where all Priority 1 and 2 checkpoints are satisfied; and Level AAA where all Priority 1, 2, and 3 checkpoints are satisfied.

2.1.2 Section 508 (US)

In 1998, the US Congress amended the US Rehabilitation Act to require from US Federal that their electronic and information technology be accessible to people with disabilities [26]. Section 508, applying to all Federal agencies when they develop, procure, maintain, or use electronic and information technology, was passed with the aim to eliminate barriers in IT, create new opportunities for people with disabilities, and to encourage the development of technologies to achieve these goals. The technical Standards part of the law provides criteria specific to various types of technologies, including software applications and operating systems, web-based information or applications, telecommunication products, video and multimedia products, self contained, closed products (e.g. information kiosks, calculators, and fax machines), and desktop and portable computers. The functional performance requirements of the law are intended for overall product evaluation and for technologies and are designed to ensure that the individual accessible components work together to create an accessible product that would 'allow people with sensory or physical disabilities to locate, identify, and operate input, control and mechanical functions and to access the information provided, including text, static or dynamic images, icons, labels, sounds or incidental operating cues'.

2.1.3 BITV (German)

The BITV (Barrierefreie Informationstechnik Verordnung (Barrier free Information Technology Ordinance)) [3] was published in July 2002 by the German Department of State, according to which the Federal authorities in Germany has made a commitment that they would launch a barrier-free version of its home records by the end of 2005. The BITV formulated requirements that a barrier-free website must meet the guidelines of WAI and WCAG 1.0. The ordinance applies to websites of authorities of the Federal Administration, and websites and graphic user interfaces which are publicly accessible. Standards are formulated for accessible web design, but they are not related to technical implementation of a barrier free web design. In December 2006 the German Federal Ministry of Labour and Social Policy announced that the BITV will be revised in accordance with WCAG 2.0. The BITV is split into two priorities. In accordance with the annex to this ordinance, IT and internet facilities shall be designed in a way that all facilities fulfil the standards and requirements listed under priority I and central navigation and start pages also take into account the standards and requirements listed under priority II.

2.1.4 Fujitsu

The Fujitsu Web Accessibility Guidelines [10] promote an easy, user-friendly experience and will be localized for each country and provided to the public as they become available. The guidelines place importance on practical effect, the potential for global application, and coordination with existing standards and guidelines, including WAI's WCAG 1.0, Section 508, and the Soumu-sho (Japan's Ministry of Public Management, Home Affairs, Posts & Telecommunications) Web Helper. There are three levels of priority: Priority 1 should be regarded as essential items for all websites, Priority 2 is strongly recommended and Priority 3 is recommended. There are also some guidelines which fall outside the subject of accessibility, but which should be considered when designing websites. The priority levels are not fully in line with that of WCAG.

2.2 Related Work

Due to space limitations we cannot include a complete review of related work, but we include some of the most prominent work that highlights issues associated with the arguments in this paper. Comparisons were performed between various automatic tools with regards to conformance of a website to accessibility guidelines, e.g. between Bobby and LIFT Machine [12]. Drawbacks of automatic accessibility tools have been highlighted such as: long and detailed results that are difficult to interpret; detection of only a limited number of barriers and manual inspection is still required; developers need to understand the requirements of the various guidelines; usability is not tested together with accessibility; lack of recommendations that are easy to understand [24]. Similarly, problems with the WCAG guidelines themselves have been pointed out, e.g. Kelly et al. [18] highlighting issues such as: dependencies on other WAI guidelines; the theoretical nature and complexity of the guidelines and the level of understanding of accessibility issues required when using the guidelines. Even though they do not illustrate their claims with examples or explain how they came to these conclusions, their list of drawbacks supports our findings that are highlighted in section 4.2. The importance of creating tools to assist developers in ensuring that their websites are accessible can be seen in the establishment of the IRIS European Project [16]. One component of the project, EvalIris Web Service [22], was

created to check the accessibility of web pages against the WCAG 1.0 guidelines. Another initiative, the European Accessibility Observatory (EIAO) [6] project, has developed an observatory to perform large scale automatic accessibility benchmarking according to the Unified Web Evaluation Methodology (UWEM) of European websites. Furthermore, The EU Web Accessibility Benchmarking Cluster, Evaluation and benchmarking of Accessibility (WAB Cluster) [29] has been established. The goal of the WAB Cluster is to cluster a number of European projects to develop an European methodology to evaluate and benchmark the accessibility of websites. This Unified Web Evaluation Methodology (UWEM) only addresses evaluating the accessibility of a website with an expert and using automatic tools, and does not address user testing. However, in a developing country context, as mentioned in the Introduction, most of the time an accessibility expert is not available.

3. Methodology for Accessibility Evaluation of Interactive Systems

We used the National Accessibility Portal (NAP) [23] as case study to develop and refine the methodology and to illustrate the use of automatic accessibility tools to draw conclusions about the value and usefulness of their application. NAP was specifically designed for people with a variety of disabilities in mind.

The feedback obtained from user testing (refer to Section 4.2) made us realise that it will not be possible to test for all problems using only automatic tools, since human judgement is sometimes required.

Since the luxury of a full-blown usability evaluation is not possible in most contexts, as described in the Introduction, it was necessary to develop a lightweight methodology that can easily be applied by the developers of similar system themselves. This involved the use of a combination of off-the-shelf automatic accessibility tools and user testing.

Two further versions of NAP were used to test the methodology. Some of the results of version 3 are discussed in Section 4.

The proposed methodology consists of 3 iterative phases:

1. *Initial accessibility evaluation:* The first phase is to assess the accessibility of an interactive system before it is tested by users. The free automatic tools selected must address at least the following aspects of accessibility:
    a. Whether the website adheres to a specific set(s) of guidelines;
    b. The readability of the text that is used in the website to ensure that the text is not too difficult to understand for the intended users;
    c. The colour-contrast of the pages of the website to ensure that visually impaired persons can still use the website; and
    d. Whether a website is accessible for persons with visual impairments who make use of a screen reader.

    Any errors found should be fixed and the tests rerun before going onto the next step of user testing. Examples of possible free tools that are appropriate to assess these aspects of accessibility are listed in Table 1 (and discussed in section 4).
2. *User testing with real users:* Accessibility issues that are not captured in the relevant set of guidelines addressed by a specific automatic tool will not be highlighted by that tool. In addition, to ensure both accessibility and user acceptance of an interactive system, it is important to perform user testing (not to be confused with usability testing) of the system with participants that are representative of typical users of the system, in this case people with various disabilities.
3. *Develop in-house 'guidelines' relevant to the context:* Based on the 'lessons learned' during the user testing and the results of using the automatic tools, develop a set of context specific in-house 'guidelines' that can be easily understood by non-expert developers. These 'guidelines' can be used to inform the design, development and testing of future versions of the system to prevent the repetition of the same mistakes. These are informal in-house 'guidelines' and not heuristics or standards.

Table 1. Automatic tools used to check accessibility of web page

Accessibility Aspect	Tool	Comment
Adherence to guidelines; recommended for people with limited programming knowledge	TAW [5]	Tests against WCAG 1.0 Guidelines.
	Cynthia Says [15]	Tests against WCAG 1.0 Guidelines and Section 508.
	Fujitsu Web Accessibility Inspector [11]	Tests against WCAG 1.0 and Fujitsu Guidelines.
	Torquemada [4]	Tests against WCAG 1.0 Guidelines.
Adherence to guidelines; recommended for people with programming experience	Total Validator [25]	Tests against WCAG 1.0 Guidelines and Section 508.
	Imergo®Online [8]	Tests against WCAG 1.0 Guidelines, Section 508 and BITV.
Readability	Readability Index Calculator [21]	Cannot evaluate a whole webpage, but have to copy and paste relevant text into text area supplied on their website.
	JuicyStudio Readability Test [17]	Tests a whole webpage -headings and links also used during the calculation of the scores. Provides breakdown of text that can assist in adapting text to improve score.
Colour Contrast	Fujitsu ColorSelector [9]	Each colour combination has to be selected manually. Didn't identify colour contrast problems that were highlighted by the other two tools.
	JuicyStudio Colour Contrast Analyser Firefox Extension [17]	Checks complete webpage - checks luminosity contrast ratio, colour difference and brightness difference of elements specified in the DOM.
	AccessKeys AccessColor [2]	Checks complete webpage - checks colour contrast and colour brightness of elements specified in the DOM. Also provides link to specific line in the code where the problem occurs.
Screen reader emulator	Fangs [20]	Provides textual representation of web page, similar to how page will be read by screen reader. Also provides list of headers and links.

4. Case Study

In this section we discuss each of the three iterative phases of the methodology using NAP as case study. Section 4.1 discusses the

results obtained using automatic accessibility tools to test NAP against sets of accessibility guidelines. User testing is discussed in Section 4.2 and the development of in-house 'guidelines' is discussed in Section 4.3.

4.1 Initial Accessibility Evaluation of NAP

Since most of the automatic tools test one webpage at a time and cannot test an entire website in one go, a representative set of the various types of pages on the portal were selected for testing and refinement of the methodology over the three versions of the portal. Section 4.1.1 discusses the results obtained using tools focusing on testing the application against sets of guidelines. Section 4.1.2 focuses on the readability results when the pages were tested with an automatic tool, while the colour contrast results of specific pages of the portal are presented in Section 4.1.3. Section 4.1.4 addresses the use of screen reader emulators to simulate the output of screen readers. Section 4.1.5 concludes with some general comments and observations.

4.1.1 Accessibility Evaluation of NAP Portal against Sets of Guidelines

To enable us to compare results and to investigate whether the same results are obtained when different tools are used, we used various automatic tools that test the accessibility of websites against established sets of guidelines. Table 2 lists the various tools used with the guideline sets addressed by the specific tool. The results described were obtained applying the tools to the third version of NAP.

Table 2. Automatic tools used to check accessibility of web page against sets of guidelines

Tools	**Guidelines**			
	WCAG 1.0	Section 508	BITV	Fujitsu
Fujitsu Web Accessibility Inspector	X			X
HiSoftware Cynthia Says	X	X		
Imergo® Online	X	X	X	
TAW	X			
Torquemada	X			
Total Validator	X	X		

4.1.1.1 Fujitsu Web Accessibility Inspector

Fujitsu Web Accessibility Inspector [11] evaluates the accessibility of a webpage against various priority levels of the WCAG 1.0 guidelines, as well as various priority levels of Fujitsu guidelines. A standalone application can be downloaded and testing is done per webpage. The report provides a description of the problem, the priority level and whether it is definitely a problem or only a warning, the possibility to view the item visually in a separate window, a link to the line in the code and a link to the guideline where a more detailed explanation of the guideline is provided. The following errors were, for example, found with regards to the WCAG 1.0 and Fujitsu guidelines:

- JavaScript is used, but no alternative is given for cases where users disable JavaScript (WCAG Level 1, Fujitsu Level 2).
- There is insufficient contrast between the text colour and the background colour (WCAG Level 3, Fujitsu Level 1).
- Many functions depend on a mouse click. However, the user should be able to navigate the page with a keyboard (WCAG Level 1, Fujitsu Level 1). Many users with mobility impairments (limited motor control) and visual impairments make use of the keyboard to navigate and not the mouse.
- More descriptive words should be used for links, e.g. 'If you forgot your password, click here' (WCAG Level 2, Fujitsu Priority 2). Many browsers provide the functionality to read aloud text that is linked. However, if all links are indicated by here or click here, it will be difficult for the user to distinguish between the various links.

An observation made was that Fujitsu has in fact identified different priority levels for some of the WCAG Guidelines when they created their own set of guidelines.

4.1.1.2 Cynthia Says

HiSoftware Cynthia Says [15] validates the accessibility of web content and identifies errors according to the US Section 508 standard and the WCAG 1.0 guidelines. The testing is done by entering the webpage URL online. The report is very textual, where the guideline is explained in a lot of detail (description of guideline and all rules that apply to the guideline) and whether the page passed the test for the specific guideline or not. The following errors were identified:

- Metadata should be provided to add semantic information to pages, such as description, keywords and language (WCAG Level 2).
- Avoid using deprecated features of W3C technologies, e.g. the element *img* with the deprecated attribute align (WCAG Level 3).
- The same words should not be used for different links, e.g. 'To register as a new user, click here' (WCAG Level 2).
- The primary natural language of the page should be identified (WCAG Level 3).

With this tool no errors were identified with regards to the Section 508 guidelines.

4.1.1.3 Imergo® Online

Imergo® Online [8] examines the accessibility of a webpage according to the WCAG 1.0 guidelines, the BITV and Section 508 standards. The report indicates the guideline (with the guideline number and its priority level) and whether it is only a warning, or an error, and displays the applicable line of code where the error occurred. The following errors were, for example, identified for WCAG 1.0 and BITV:

- Keyboard alternatives should be provided for mouse click events (BITV Level 1, WCAG Level2).
- Deprecated attributes should not be used (BITV Level 1, WCAG Level 2).
- Alternatives should be provided where JavaScript is used (BITV Level 1, WCAG Level 1).
- Text should be provided as an alternative to an image (WCAG Level 2).
- The primary language of the document should be indicated (WCAG Level 3).

4.1.1.4 TAW

TAW (Web Accessibility Tool) [5] assesses the webpage accessibility based on the WCAG 1.0 guidelines. The URL is entered online and the priority level can be selected. It illustrates, visually overlaying the website, where the issues/problems occur

and then describes it in more detail. It clearly indicates errors and possible errors where human intervention is required to judge whether it is an error or not, and provides a summary of the number of errors per priority level that could be identified automatically, and the number of possible errors that should be judged by a human. The following errors were, for example, identified with regards to WCAG 1.0:

- Alternatives should be provided to dynamic content using JavaScript (WCAG Level 1).
- The user should be able to read the page without style sheets (WCAG Level 1).
- Headers should be properly nested, i.e. header levels should not increase by more than one level per heading (WCAG Level 2).
- Deprecated features should be avoided (WCAG Level 2).
- There should be efficient contrast between the text and background colour (WCAG Level 2).
- The target of links should be clearly identified (WCAG Level 2).
- Summaries of tables should be provided (WCAG Level 3).

4.1.1.5 Torquemada

Torquemada [4] verifies the accessibility of the webpage according to the WCAG 1.0 guidelines. No priority levels of the guidelines are selected. The results are reported graphically or textually, depending on the user's selection. The report page is divided into three frames: the errors or warning with regards to the guidelines, together with the element and attribute, the number of instances and the code lines where it occurs; the website how it is displayed in a browser; and the source code. The following errors were, for example, identified by the tool with regards to WCAG 1.0:

- A logical tab order should be specified. This is very important for people with visual impairments that use a screen reader.
- The page should be correctly displayed when style sheets are disabled.
- An alternative should be provided so that the page works correctly when JavaScript is disabled.
- The target of a link should be clearly described.
- Alternatives should be provided to mouse events – if a user navigates the site with a keyboard, appropriate alternatives should be available.
- Summaries should be provided of information contained in tables.

4.1.1.6 Total Validator

Total Validator [25] validates the HTML, accessibility, spelling and broken links. The web accessibility validation is done against the WCAG 1.0 guidelines and the US Section 508 standard. A stand-alone application, as well as a Firefox extension, is available. The report contains the code and the errors or warnings with regards to the guidelines are displayed in bold, along with the error code, within the source code where it occurs. The following errors were, for example, identified for WCAG 1.0 and Section 508:

- An alternative should be provided for cases where the user does not activate JavaScript (WCAG Level 1, Section 508).
- Headings should be ordered properly – headings should not increase more than one level, e.g. in HTML H2 should follow H1, etc (WCAG Level 2).
- Similar text should not be used for different links (WCAG Level 2).
- The primary language of the document should be identified (WCAG Level 3).

4.1.1.7 Overall

It is interesting to note that these tools do not always identify the same errors and one can therefore not completely rely on a single tool claiming to do the same as another (see Table 3 for summary of the various errors and which tools identified which errors).

Table 3. Automatic tools used to check accessibility of web page and the errors that they have identified

	Errors					
Tools	Fujitsu Web Accessability	HiSoftware Cynthia Says	Imergo® On-line	TAW	Torquemada	Total Validator
JavaScript Alternative	X		X	X	X	X
Image Alternative			X			
Style sheet Alternative				X	X	
Mouse Alternative	X		X		X	
Header Nesting				X		X
Meta data		X				X
Deprecated Attributes		X	X	X		
Colour Contrast	X			X		
Descriptive Links	X	X				X
Summaries of tables				X	X	
Logical tab order					X	

Even though these automatic tools test whether a website adheres to sets of guidelines, this is not enough to ensure accessibility. For persons with intellectual disabilities or who are not as literate, the text of the website should be easy to read and understand. Some automatic tools that can be used to test the readability of text are discussed in section 4.1.2. Tools that test the contrasts of colours used in a webpage are discussed in 4.1.3. Furthermore, even if a website adheres to the accessibility guidelines, it is not necessarily accessible to people that make use of a screen reader. To check what output is given to a screen reader, as well as the headings and links, a screen reader emulator can be used. An example of such an emulator is discussed in section 4.1.4.

4.1.2 Evaluation of the Readability of the Information on the NAP Portal

In order to test the readability of the NAP portal the Readability Index Calculator [21] and JuicyStudio Readability Test [17, 19] were used. These tools calculate a readability index score for the selected text using the Flesch-Kincaid reading and grade levels [7, 19]. The Flesch-Kincaid Reading Ease score indicates how easy a text is to read and a high score implies an easy text, e.g. comics typically score around 90 while legal text can get a score below 10. The Flesch-Kincaid Grade level indicates the grade a person

should have reached in order to be able to understand the text, e.g. a grade level of 7 indicates that a seventh grader will be able to understand the text. It should be noted, that these tools only provide a rough indication of the readability of the text, since they tend to reward short sentences made up of short words.

Unfortunately with the Readability Index Calculator you cannot evaluate a whole webpage, but have to copy and paste the relevant text into the text area supplied on their website, in order to evaluate it. The results only indicate the Flesh-Kincaid grade level and the Flesch-Kincaid Reading ease score.

With the JuicyStudio Readability Test you can test a webpage by typing in the URL. The results indicate the Flesh-Kincaid grade level and the Flesch-Kincaid Reading ease score, as well as a breakdown of the text as illustrated in Figure 1.

A reading ease score of 90.0–100.0 indicates text that easily understandable by an average 11-year old student; a score of 60.0–70.0 text that easily understandable by 13- to 15-year old students, and a score of 0.0–30.0 text best understood by college graduates. A grade level score corresponds with a typical school grade level. For example, a score of 8.2 would indicate that the text is expected to be understandable by an average student in 8th grade (usually around ages 13-14) [7, 19].

The 'About' page of the NAP portal was tested with these two tools. Interestingly, they didn't present the same results. According to the Readability Index Calculator, the Flesch-Kincaid Reading Ease Score is 22 and the Flesch-Kincaid Grade Level is 14. However, according to the JuicyStudio Readability Test, the Flesch-Kincaid Reading Ease Score is 44.62 and the Flesch-Kincaid Grade Level is 9.21.

This indicates that the values obtained will depend on the tool that is being used. Since the JuicyStudio Readability Test tests a whole webpage, headings and links are also used during the calculation of the scores, and this can impact on the score levels that are obtained.

These tools can assist in establishing whether the text is appropriate for the intended audience. However, if the text has to be adapted, that can only be done by a human.

Reading Level Results

Summary	Value
Total sentences	65
Total words	570
Average words per Sentence	8.77
Words with 1 Syllable	325
Words with 2 Syllables	95
Words with 3 Syllables	82
Words with 4 or more Syllables	68
Percentage of word with three or more syllables	26.32%
Average Syllables per Word	1.81
Gunning Fog Index	14.03
Flesch Reading Ease	44.62
Flesch-Kincaid Grade	9.21

Figure 1. A breakdown of the text that is presented in the result report of the JuicyStudio Readability Test

4.1.3 Evaluation of the Colour Contrast of the Pages of the NAP Portal

Fujitsu ColorSelector [9], JuicyStudio Colour Contrast Analyser Firefox Extension [17] and AccessKeys AccessColor [2] were used for evaluating the colour contrasts used in NAP.

Fujitsu ColorSelector is a standalone application that can evaluate whether the selected colour combination for the text and background are easily viewed by people with cataracts or colour blindness. The user can either fill in the RGB values of the colours or use the dropper to select the text colour and background colour from the website. Each combination has to be selected manually and the user cannot simply provide the URL or a webpage to be evaluated (refer to Figure 2).

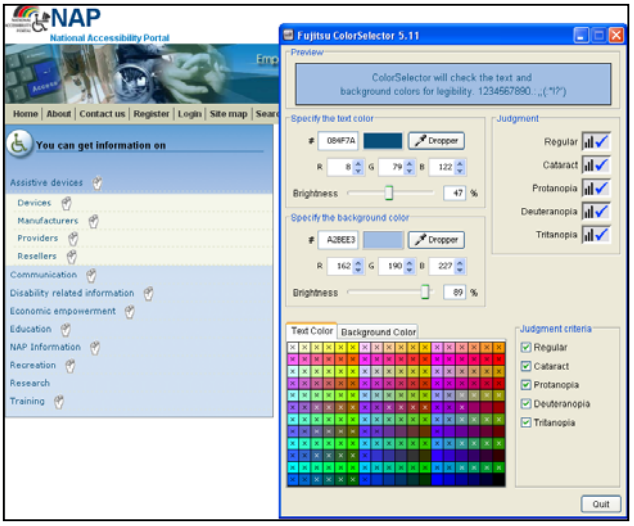

Figure 2. Fujitsu ColorSelector where the ticks in the boxes on the right hand side indicate that the specific colour combination has passed all tests.

JuicyStudio Colour Contrast Analyser Firefox Extension enables the user to check the colour contrast of elements specified in the DOM. When a user wants to test a webpage this tool will save them time and the user does not have to guess which combination to test. It evaluates whether the luminosity contrast ratio, colour difference and brightness difference is above the threshold according to the algorithm suggested in the 26th of April 2000 working draft for Accessibility Evaluation and Repair Tools (AERT) [28]. The tool lists colour combinations used in the document in a table that summarises the foreground colour, background colour, luminosity contrast ratio, colour difference and brightness difference. Any problems are highlighted in yellow in the report. The report also provides a summary of failures, as well as the thresholds that have been used for the evaluation (refer to Figure 3).

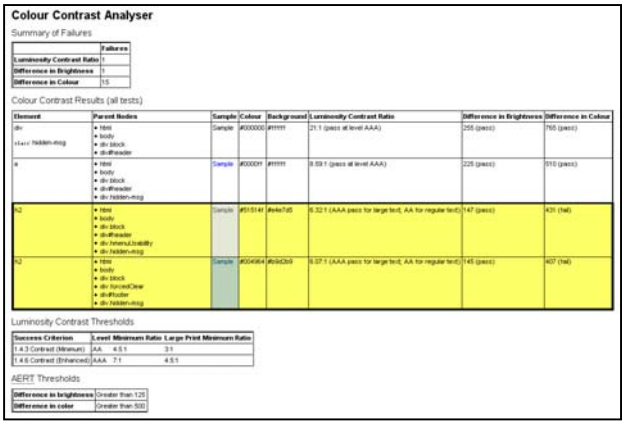

Figure 3. JuicyStudio Colour Contrast Analyser where the combinations that failed the test are highlighted in yellow. On top is a summary of the failures and below a summary of the thresholds that were used for the tests.

AccessKeys' AccessColor tests whether the colour contrast and colour brightness between the foreground and background of all elements in the DOM are high enough for people with visual impairments. The user types in the URL of the webpage and then a report is generated. The report provides a summary of the failures and presents the thresholds that were used for the tests. It indicates any failures with a red symbol. The code is displayed below the results and each element has a link to the specific line of code. This is illustrated in Figure 4.

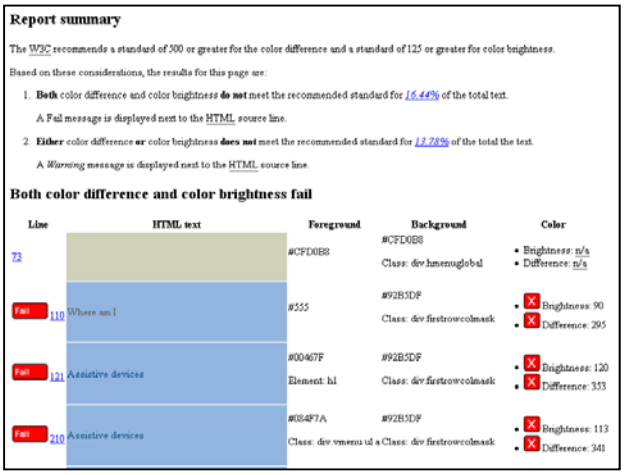

Figure 4. AccessKeys AccessColour report where the combinations that failed the test indicated in red. On top is a summary of the failures.

Fujitsu ColorSelector can be time-consuming when a whole website or webpage has to be tested and no specific combination(s) has been identified for testing. It also failed to identify a possible problem where blue text was used on a blue background as illustrated in Figure 2. According to this tool, it passed all the tests. However, when checked by a human, a possible problem is clearly identified. This is confirmed by the results of both JuicyStudio Colour Contrast Analyser and AccessKeys AccessColor. Both of these tools identified this specific colour combination to be problematic. The same result was also found with many other colour combinations, where the Fujitsu ColorSelector didn't indicate any problem, but the other tools either failed this combination on the brightness difference, colour difference or both.

JuicyStudio Colour Contrast Analyser also tests for the luminosity contrast ratio, but this is not tested by AccessKeys AccessColor. However, AccessKeys AccessColor provides a link to the specific line in the code where the problem occurs. This is not provided by JuicyStudio Colour Contrast Analyser. According to these two tools, many colour combinations for the text and background throughout various pages of the NAP portal failed.

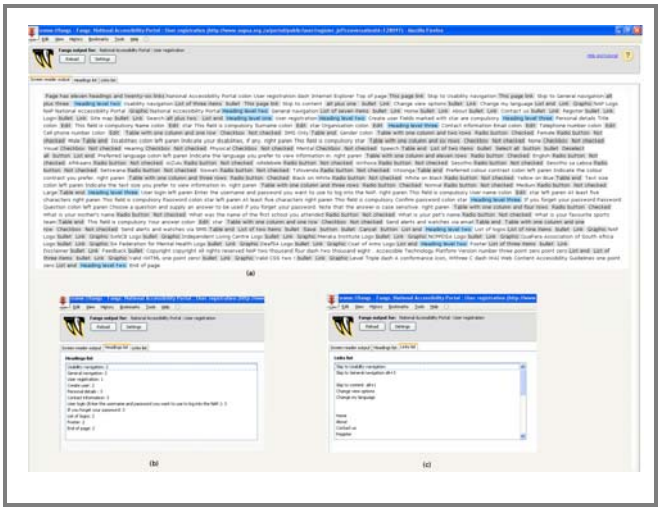

Figure 5. Output from Fangs for a webpage. (a) Screen reader output, representative of how a screen reader will read the webpage (b) Header list (c) Link list

4.1.4 Evaluating with Screen Reader Emulator

When a screen reader, such as JAWS, is used by someone to test the website, it can be quite difficult to identify possible problems. This section discusses an alternative option to check whether a website is accessible for people with visual impairments who use a screen reader. The synthetic voice is difficult to follow when you are not used to listening to it. However, this problem can be solved when a tool such as Fangs [20] is used.

Fangs is a screen reader emulator that provides a textual representation of the web page, similar to how a page will be read by a screen reader. It also provides a list of headers and a list of links, similar to the list that will be provided by the screen reader to the user when s/he selects this functionality. This is illustrated in Figure 5.

When a screen reader emulator, such as Fangs, is used, the following possible errors can be checked and rectified, if present:

- Whether the tabbing between elements are in a logical order.
- Whether each link has a unique description and not something like 'here' or 'click here'. Whether all headers are descriptive enough.
- Whether all headers are nested correctly.

4.1.5 General Comments and Observations

This section discusses observations that were made when testing the accessibility NAP portal with automatic tools. It presents

observations made with regards to the knowledge or experience of developers or designers and the automatic tools themselves.

4.1.5.1 Knowledge and experience of developers and designers

Many developers or designers have a lack of knowledge in terms of designing accessible websites. Therefore, when they are required to implement the accessibility guidelines, they do not always understand exactly how to achieve the desired results. In the WCAG 1.0 guidelines, no examples are given to explain to someone that is new to the field what is meant by each guideline. This can lead to many different interpretations of what is required. Therefore, examples and tools are required to assist developers or designers to overcome these obstacles.

4.1.5.2 Automatic tools

There are a vast variety of tools available and choosing the most appropriate tools, without any experience of using these tools before, can be a daunting task, especially when working under time pressure. Furthermore, when automatic tools are used to test the accessibility of a website, many tool reports are very technical and not easy to understand, especially for a designer that does not have a very technical background. In many cases the tool identifies a possible problem, but highlights that a human should make sure whether it is an actual problem or not. For a person with limited knowledge and experience in accessible design, this is not trivial to do.

Table 4. Automatic tools used to check the code against sets of guidelines.

Tools	**Code Validation**				**Guidelines**
	CSS	HTML	XHTML	XML	
Imergo® Online	X	X	X	X	WCAG 1.0 Section 508 BITV
TAW		X	X		WCAG 1.0
Torquemada	X	X	X		WCAG 1.0
Total Validator		X	X		WCAG 1.0 Section 508
W3C CSS Validation Service [27]	X		X		W3C Markup

Based on accessibility tests that were performed with the automatic tools on the NAP portal, as well as a user testing of the NAP portal [13], the following limitations of automatic tools can be identified:

- Many of the tools check only static pages and if your pages include dynamic content, it becomes difficult to automatically test the accessibility with automatic tools. If video content is displayed on the website, the tools will not test the accessibility of the video.
- There are many tools that validate the code against certain guidelines (see Table 4). However, when people with limited or no knowledge of the code, such as graphic designers, have to evaluate the accessibility of a website with automatic tools, they require a report that is easy to understand and not too technical. From the above mentioned tools, we recommend TAW, Cynthia Says or Fujitsu Web Accessibility Inspector for people with limited programming knowledge and Torquemada, Total Validator or Imergo®Online for people with programming experience.
- Automatic tools cannot test whether the placement of certain elements, e.g. headings, affects the accessibility of the page. Even if the contrast between the text and the background is sufficient, but the placement is chosen poorly, it will still not be visible to the users and can lead to frustration.
- Automatic tools do not always provide the functionality to check how the website is displayed in various browsers. Even if this functionality exists, a human would have to check whether the page is displayed correctly in the various browsers, e.g. Mozilla Firefox and Internet Explorer.

4.2 User Testing

Automatic tools will not pick up errors where human judgement is required, for example the placement of specific elements on the page. Therefore, user testing is required to identify these kinds of issues.

Expert usability reviews and full-blown usability testing tend to find high level breaches of design rules and consistency, but may overlook some potentially serious ones related to the special domain and the use of various assistive technologies. Following accessibility guidelines and good practices alone therefore cannot ensure that the website is truthfully accessible. In the absence of a usability expert, the automatic tools might go a long way to find adherence to the standard guideline, but user testing with 'real users' would be an essential additional requirements.

Three general types of user testing for accessibility can be identified [1]:

1. 'Home' user testing, where the user uses the site in his/her own context (work or home) and where applicable, using the assistive devices s/he is used to without direct 'interference' of a third person. The user can be asked for free response on his/her experience with the website, or can be interviewed by the designer (or even developer) of the site to get feedback.
2. Moderated user testing, possibly including some screen capturing or video recording (that can be analysed later), where the user discusses the website based on moderator prompts whilst using it.
3. Full-blown lab testing with the involvement of usability experts.

Home testing is the most desirable and we have found probably the most practical (due to the difficulty that disabled users often have to travel to 'formal' testing sites).

NAP was designed with screen readers in mind and tested with Canoo [31] and Bobby [30] during development of the first version. However, when the first version of NAP was tested by disabled users during an earlier user testing [13] the following examples of typical problems, not picked up by these automatic tools, were identified by the mobility impaired and visually impaired users:

- When choices are presented through checkboxes, there should always be an option to select all the checkboxes. For persons with limited motor control, it takes a lot of effort to select all the various check boxes individually.
- Users with mobility impairments struggle to move the mouse or cursor with precision. Therefore, when radio-buttons are

used, the selection area should be resizable to a large size so that the user can select the radio-button with less effort.

- The first field of a form should automatically be activated to reduce the effort of the user to first select the field before s/he can start to enter his/her input.

4.3 Developing In-house 'Guidelines'

Lessons learned from the evaluation of the system with the automatic accessibility evaluation tools, as well as the user testing, are used to create in-house 'guidelines' to ensure that the same mistakes are not made in future versions of the system.

Sections 4.3.1 to 4.3.5 give examples of in-house 'guidelines' that were developed as a result of the applying the methodology on NAP. These guidelines are in fact generic in nature and should be applied to any website aiming for a reasonable level of accessibility.

4.3.1 Navigation

- Alternatives should be provided to dynamic content using JavaScript.
- Navigation should not only depend on using the mouse, but should cater for users that only make use of a keyboard to navigate, i.e. keyboard alternatives should be provided for mouse click events.
- Each link should have a unique description that is descriptive enough that the user understands exactly what the link is for, i.e. target of links should be clearly identified.
- A logical tab order should be specified. This is very important for people with visual impairments that use a screen reader.

4.3.2 Look and Feel

- The selected colours for text and the background should have sufficient contrast.

4.3.3 Code

- Metadata should be provided to add semantic information to pages, such as description, keywords and language.
- Deprecated features of W3C technologies should be avoided, e.g. the element *img* with the deprecated attribute align.
- The user should be able to read the page without style sheets.
- Headers should be properly nested, i.e. header levels should not increase by more than one level per heading.

4.3.4 Information

- Text should be provided as an alternative to an image.
- Text summaries of tables should be provided.

4.3.5 Selection

- When choices are presented through checkboxes, there should always be an option to select all the checkboxes.
- When radio-buttons are used, the selection area should be resizable to a large size so that the user can select the radio-button with less effort.
- The first field of a form should automatically be activated to reduce the effort of the user to first select the field before s/he can start to enter his/her input.

5. Conclusion

Accessibility experts are mostly not available in the developing world context. When such experts are not available, reliance on automatic tools combined with some user testing, might be the only way to go. Developers and designers cannot, however, rely on a single automatic tool, or on automatic tools only, to check the accessibility of their website. Even though these generic tools will go some way in indentifying crude accessibility issues, they will tle>Journal of Applied Psychology</full-ges>221-233</

Each person with a disability has his/her own way of interacting with ICT devices and even prefers to use different assistive devices, depending on their computer literacy and the severity of their disability. Users testing in the context of the users' own environments are therefore important and desirable.

The methodology we proposed to improve the accessibility of an interactive system, as illustrated in Figure 6, was developed and refined over three versions of developing NAP. The methodology has three iterative steps, namely: evaluating accessibility of websites with automatic accessibility tools; user testing; and developing in-house 'guidelines' based on lessons learned from the first two phases to ensure that the same mistakes are avoided in future. We demonstrated the three phases with a case study, using NAP that was developed with persons with disabilities in mind.

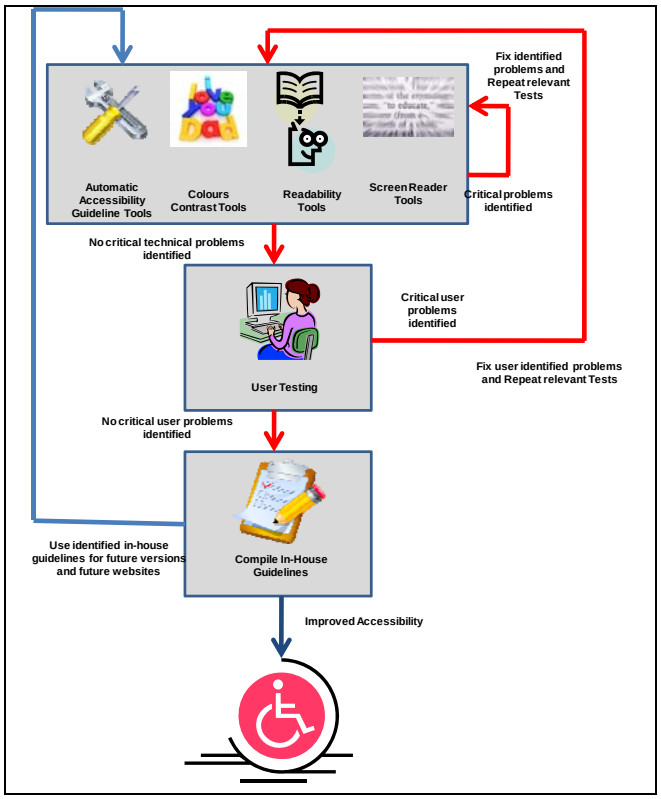

Figure 6. Lightweight methodology to improve web accessibility

6. REFERENCES

[1] AbilityNet: *Disabled user testing*. (2009) [cited 20 June 2009]; Available from: http://www.abilitynet.org.uk/webtesting.

[2] AccessKeys: *AccessColor*. (2009) [cited 29 January 2009]; Available from: www.accesskeys.org/tools/color-contrast.html.

[3] Aktion Mensch: *Federal Ordinance on Barrier-Free Information Technology*. (2002) [cited 29 January 2009]; Available from: http://www.einfach-fuer-alle.de/artikel/bitv_english/.

[4] Bordoni, F.U.: *Torquemada*. (2009) [cited 29 January 2009]; Available from: www.webxtutti.it/testa_en.htm.

[5] CTIC Foundation: *TAW Web Accessibility Test*. (2009) [cited 29 January 2009]; Available from: www.tawdis.net/taw3/cms/en.

[6] EIAO: *European Internet Accessibility Observatory*. (2009) [cited 29 January 2009)]; Available from: http://www.eiao.net.

[7] Flesch, R.: A new readability yardstick. *Journal of Applied Psychology*. 32 (1948), 221-233.

[8] Fraunhofer: *Imergo® Online*. (2009) [cited 29 January 2009]; Available from: http://imergo.com/home.

[9] Fujitsu: *Fujitsu Color Selector*. (2009) [cited 29 January 2009]; Available from: www.fujitsu.com/global/accessibility/assistance/cs/.

[10] Fujitsu: *Fujitsu Web Accessibility Guidelines*. (2009) [cited 29 January 2009]; Available from: http://www.fujitsu.com/global/webaccessibility/.

[11] Fujitsu: *Fujitsu Web Accessibility Inspector*. (2009) [cited 29 January 2009]; Available from: www.fujitsu.com/global/accessibility/assistance/wi/.

[12] Giorgio, B.: Comparing accessibility evaluation tools: a method for tool effectiveness. *Universal Access in the Information Society*. 3(3-4) (2004), 252-263.

[13] Greeff, M.,Kotze, P., *Individual difficulties faced by persons with mobility impairments*. In: (eds.): Proceedings of IADIS: International Conference on Interfaces and Human Computer Interaction. IADIS, Lisbon, 2007.

[14] Greeff, M.,Kotze, P., *I am Part of Society, but Still an Individual: A Case Study about Challenges Faced by Individuals with Mobility Impairments*. In: (eds.): Proceedings of Accessible Design in the Digital World: New Media; New Technologies; New Users. University of York, 2008.

[15] HiSoftware: *Cynthia says*. (2009) [cited 29 January 2009]; Available from: www.contentquality.com/.

[16] IRIS: *IRIS Project Website*. (2009) [cited 29 January 2009]; Available from: http://www.iris-design4all.org.

[17] Juicy Studio: *Color Contrast Firefox Extension*. (2009) [cited 29 January 2009]; Available from: http://juicystudio.com/article/colour-contrast-analyser-firefox-extension.php.

[18] Kelly, B., Sloan, D., Phipps, L., Petrie, H., Hamilton, F., *Forcing standardization or accommodating diversity? A framework for applying WCAG in the real world*. In: (eds.): Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A). 2005.

[19] Kincaid, J.P., Fishburne, R.P., Rogers, R.L., Chissom, B.S.: *Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel*, In *Research Branch Report 8-75*. 1975, Naval Technical Training, U. S. Naval Air Station, Memphis, TN: Millington, TN.

[20] Krantz, P.: *Fangs*. (2009) [cited 29 January 2009]; Available from: http://www.standards-schmandards.com/projects/fangs/.

[21] Krantz, P.: *Readability Index Calculator*. (2009) [cited 29 January 2009]; Available from: www.standardsschmandards.com/exhibits/rix/index.php.

[22] LIPCNE: *EvalIris - A Web Service for Web Accessibility Evaluation*. (2009) [cited; Available from: www.sc.ehu.es/acwbbpke/evaliris.html.

[23] Meraka Institute: *National Accessibility Portal*. (2009) [cited 29 January 2009]; Available from: www.napsa.org.za/portal.

[24] Rowan, M., Gregor, P., Sloan, P., Booth, P., *Evaluating web resources for disability access*. In: (eds.): Proceedings of the Fourth International ACM Conference on Assistive Technologies (ASSETS'00). ACM, 2000, 80 - 84.

[25] Total Validator: *Total Validator*. (2009) [cited 29 Janaury 2009]; Available from: www.totalvalidator.com/.

[26] U.S. General Services Administration: *Section 508*. (2009) [cited 28 January 2009]; Available from: http://www.section508.gov/.

[27] W3C Consortium: *CSS Validation Service*. (2009) [cited 29 January 2009]; Available from: http://jigsaw.w3.org/css-validator/.

[28] W3C Consortium: *World Wide Web Consortium (W3C)*. (2009) [cited 28 January 2009]; Available from: http://www.w3.org/.

[29] WAB Cluster: *Unified Web Evaluation Methodology (UWEM)*. (2008) [cited 20 June 2009]; Available from: http://www.wabcluster.org/uwem1_2/.

[30] Watchfire: *Bobby*. (2007) [cited 4 May 2007]; Available from: http://www.watchfire.com/products/webxm/bobby.apx.

[31] Webtest: *Canoo*. (2009) [cited 4 May 2007]; Available from: http://webtest.canoo.com/webtest/manual/WebTestHome.html.