

# Automatic Adaptation Techniques to Increase the Web Accessibility for Blind Users

Mexhid Ferati<sup>1</sup> and Lirim Sulejmani<sup>2</sup>

<sup>1</sup>Oslo and Akershus University College of Applied Sciences, Oslo,  
Norway  
mexhid.ferati@hioa.no

<sup>2</sup>South East European University, Skopje, Macedonia  
ls20522@seeu.edu.mk

**Abstract.** Despite the available guidelines and tools to build accessible websites, many still fail to be accessible and usable by the blind users. Web content adaptation has been used as an approach to enhance website accessibility by applying automatic transformation techniques. We describe here a system that automatically increases webpage accessibility by applying three different techniques: link enrichment, image enrichment, and navigation enrichment. Preliminary evaluation of these techniques reveals that the prototype successfully eliminates half of the accessibility errors identified by validating tools and it performs equally well regardless of the accessibility compliance level of the website.

**Keywords:** Web accessibility · Blind users · Content adaptation · Client-side

## 1 Introduction and Background

The Internet is becoming an increasingly important resource for education, commerce and recreation. To ensure easy access and equal opportunity to disabled people, web content should be accessible by being compliant with the requirements of the Web Content Accessibility Guidelines (WCAG). Most websites, however, do not fulfill these requirements, which causes issues when blind users try to access and navigate digital content [3]. This forces them to rely on help from others instead of being able to independently complete their tasks, which limits their active participation in the society [4].

Being aware of these difficulties, inspired from study [5], we built a middle-ware solution that automatically increases website accessibility using adaptation techniques. Our approach is an extension that installs on a web browser that transforms webpages into more accessible version. The extension initially checks the website for accessibility errors, and then depending on the errors found, it applies one or more of the following techniques: link enrichment, image enrichment, and navigation enrichment technique.

Preliminary evaluation of these techniques reveals that the prototype successfully eliminates half of the accessibility errors identified by validating tools and it performs equally well regardless of the accessibility compliance level of the website.

## 2 Adaptation Approach

In order to increase website accessibility, we implemented three techniques that automatically adapt the web content to improve navigation and access to information. These techniques adapt three entities: links, images, and skipping to the main content.

### 2.1 Link Enrichment

Very often links found on websites lack appropriate labelling that makes them invisible to screen readers and consequently inaccessible to blind users. For example, the webpage shown in Fig 1 contains an image that is also a link. As identified by the accessibility validator, the link lacks a text that could be accessible to screen readers. On the left side, it is shown the webpage with one error listed related to the link, while on the right side the error is eliminated after the adaptation is applied.

This technique begins by checking all hyperlinks on the page whether they lack any text or use a generic description such as ‘read more’, ‘learn more’ or ‘click here’. If such hyperlink is found, then it checks whether it leads within that same page or to another page. In cases when the link leads within the page, the heading text of that part is included as an alternate text. If the link leads to another page, the text from the title or the heading of that page is used.

This approach provides an ARIA7 label to enable links to be visible by screen readers [1]. This will automatically add an *aria-labelledby* attribute inside an ‘a’ tag with brief content taken from the section where the link is pointing, as described above. This will help blind users get more information of the page or section and help them make decision whether to visit before engaging the link. Moreover, this will display the same text in the links list, which is useful when screen reader users browse by links.

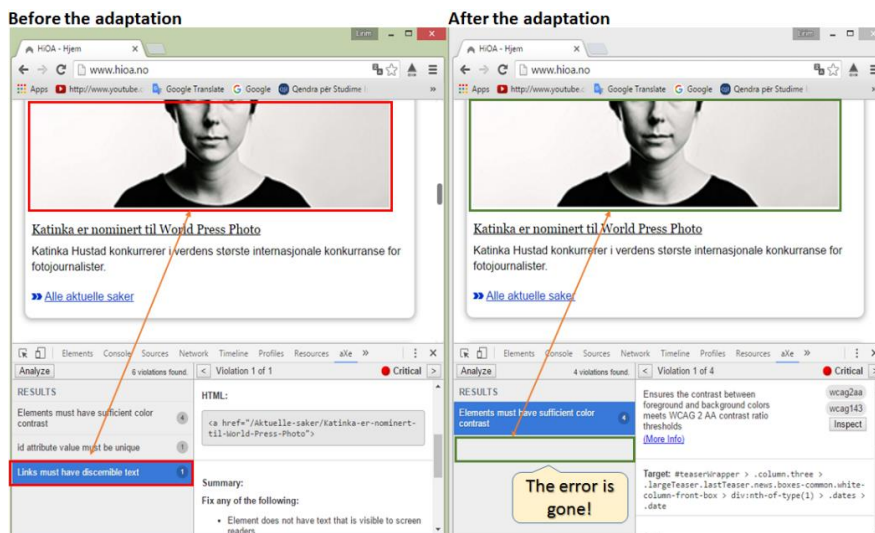


Fig 1. A before and after depiction of a link enrichment adaptation

## 2.2 Image Enrichment

Providing an alternative text to images constitutes the first principles of web accessibility comprised in Guideline 1.1 Text Alternatives from WCAG 2.0 [7]. Despite this guideline requirement, only 39.6% of images found on the homepages of the 500 most high-traffic websites, were assigned alternative text [2].

Our approach starts by identifying all images on a given page that lack an alternative text description. Once those images are identified, we initially check whether there is any text embedded in the image itself, similar to study [2]. The image is scanned using Optical Character Recognition (OCR) technique in Microsoft OneNote [8] to find any text, which is then added to the alternative tag. Very often, however, images contain no text, in which cases our approach searches for a suitable text from images on the Web.

To achieve this, we need to find similar images on Google and then use their description as a text for the alternative text. Since the Google Image Search API [9] is deprecated, we follow a work around by using the Imgur API [10]. The image lacking the alternative text is uploaded on imgur.com from where we obtain a unique URL. This URL then is sent as a parameter to look for similar images on Google Images. Once such image is found, its description then is used as a text for the alternative tag of the image. Fig 2 on the left side shows two images found without an alternate text, while on the right side those two errors are eliminated after the adaptation is applied.

## 2.3 Navigation Enrichment

This technique enables blind users to easily skip to the main content of the page. This technique is created by automatically providing a link, usually invisible and only accessible by screen readers, called skip link. By engaging the skip link, blind users can bypass on listening to the navigation menu each time they load the page and just jump

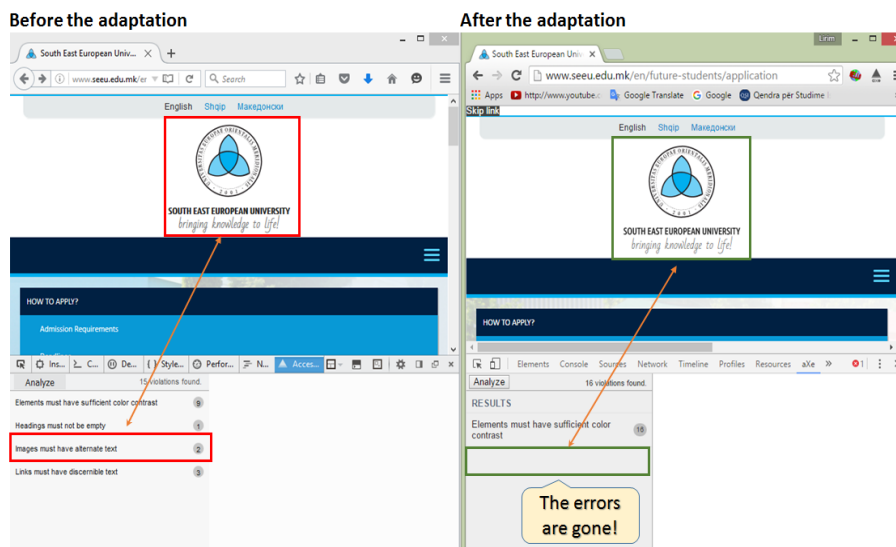


Fig. 2. A before and after depiction of an image enrichment adaptation

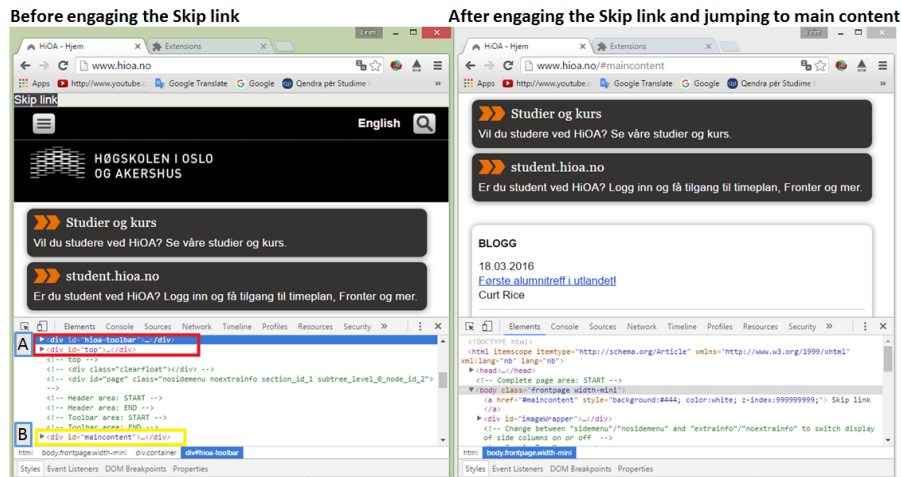


Fig 3. A before and after engaging the skip link and jumping to the page's main content

to the main content of the page [12]. Our approach will detect whether a webpage has the skip link implemented and if not, then it will be added at the top of the page. An example is depicted in Fig 3, which shows that when the provided skip link is engaged, the two sections ('hioa-toolbar' and 'top') highlighted in red (box A) are skipped and focus moves into the main content highlighted in yellow (box B).

## 2.4 Architecture and Implementation Details

The architecture and process of webpage adaptation starts by feeding the webpage into the Chrome extension (step 1), which calls the Web API (step 2). The webpage content then is gathered by the Web API and it triggers the Adaptation Plugin (step 3), which contains the adaptation knowledge base. Once the adaptation is finished, the result is sent back to the Chrome extension (step 4), which displays the webpage on the browser in an adapted, more accessible form (step 5). This process is depicted in Fig 4.

The prototype extension for this study was developed using Visual C# along with Microsoft SQL Server 2012. Additionally, JavaScript and jQuery were used along with HTML5 and CSS3. The HTML Agility Pack was used for parsing html code from the webpages. The Chrome Extension was used to make the extension for the Chrome browser, although other browser's extensions can easily be used to make the prototype deployable for other browsers. The extension source code can be located here [6].

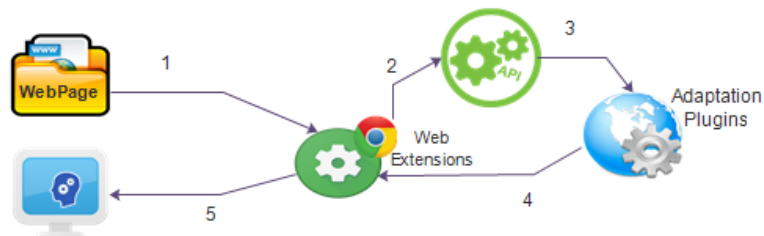


Fig 4. The architecture and the process of webpage adaptation

### 3 Evaluation

The evaluation of the adaptation techniques was conducted by two experts using three websites: hioa.no, seeu.edu.mk and bbc.com. The first two websites were chosen only because the authors come from those academic institutions while the last one is a popular news website. For each website, five random pages were independently evaluated by the experts. Initially the extension was installed into the Chrome browser and the webpage being evaluated was opened on it. For easy comparison, the same webpage was also opened on Firefox browser, which showed the original, not adapted version. Using the aXe tool [11], which is a library that does automated accessibility testing, evaluators noted the number of accessibility errors found and fixed by the extension.

#### 3.1 Findings and Discussion

Preliminary findings of the evaluation reveal that the three adaptation techniques successfully identify and fix almost all link and image related errors found by the aXe validation tool. As shown on Fig 5, the highest number of errors were fixed in the seeu.edu.mk website. Especially, the image enrichment technique had the highest impact by finding 51 errors and fixing 50 of those in ten random pages. The lowest number of errors and fixes were found in the bbc.com website, which already scored high in accessibility. Similarly, the hioa.no website had low number of errors found and fixed.

Our extension was successful in fixing 19% (78 out of 421) of total errors found by the aXe tool. This tool, however, by default also identifies errors concerning the level of color contrast on the webpages (257 errors). Since color contrast affects only partially sighted and not the blind (our target users), disregarding this type of error, increases the percentage of fixed errors to 48% (78 out of 164). In addition, considering that the lack of skip links is not identified as an error by the aXe tool, but our extension adds it when missing, the percentage of fixed errors raises to over 50%. Hence, we could claim that our techniques successfully eliminate half of all the blind user related accessibility errors found on websites. Some of the errors not addressed by our extension include: *frames must have unique title attribute*, *documents must have title element to aid in navigation*, and *form elements must have labels*.

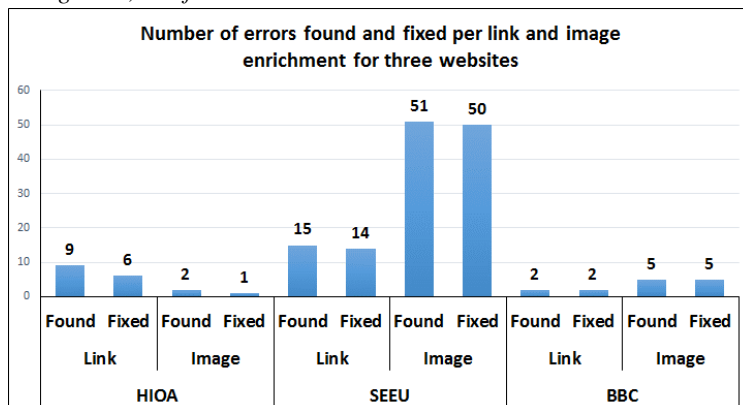


Fig 5. The SEEU website with highest number of errors found and fixed

## 4 Conclusion, Limitations and Future Work

In this paper, we presented a browser extension that automatically increases website accessibility using three adaptive techniques: link enrichment, image enrichment, and navigation enrichment. The link enrichment technique fixes links found on a website that lack or provide only generic description. The image enrichment technique locates images that lack alternative text and adds a description. Navigation enrichment technique adds a skip to main content link to every page lacking such feature.

Preliminary evaluation of these techniques with three websites showed that the implemented adaptation techniques successfully eliminate half of all accessibility errors identified by validating tools that relate to blind users. The prototype works well for all types of websites, regardless of their compliance level of accessibility, although it performs significantly slower the more images the webpage contains. For the majority of webpages we tested, it usually took between 5 and 15 seconds.

In the future, we aim to improve our tool initially to identify reasons why some of the errors identified were not fixed. Additionally, we will work on optimizing the algorithm to decrease the time requires the webpage adaptation to be performed. Finally, a more thorough and rigorous evaluation will be conducted with both experts and blind users to measure the level of appropriateness of the fixes performed.

## References

1. ARIA7 Techniques for WCAG 2.0: <https://www.w3.org/WAI/GL/2015/WD-WCAG20-TECHS-20150106/ARIA7>, Retrieved: 05.04.2016.
2. Bigham JP, Kaminsky RS, Ladner RE, Danielsson OM, Hempton GL. WebInSight: making web images accessible. Proc. of the 8th Intern. ACM SIGACCESS, pp. 181-188 (2006)
3. Di Blas N, Paolini P, Speroni M. Usable accessibility to the Web for blind users. In Proceedings of 8th ERCIM Workshop: User Interfaces for All, Vienna (2004)
4. Hollier SE. The disability divide: a study into the impact of computing and internet-related technologies on people who are blind or vision impaired. GLADNET Collection. (2007)
5. Mbipom G, Harper S. The transition from web content accessibility guidelines 1.0 to 2.0: what this means for evaluation and repair. In Proc of 27th ACM SIGDOC, pp. 37-44 (2009)
6. Extension source code: <https://github.com/lirimsulejmani/adaptivetechology>
7. Text Alternatives, Understanding Guideline 1.1, WCAG 2.0: <https://www.w3.org/TR/UNDERSTANDING-WCAG20/text-equiv.html>, Retrieved: 05.04.2016.
8. Extract text from pictures using OCR in OneNote, <https://support.office.com/en-us/article/Extract-text-from-pictures-and-file-printouts-by-using-OCR-in-OneNote-93a70a2f-ebcd-42dc-9f0b-19b09fd775b4>, Retrieved: 05.04.2016.
9. Google Image Search API, <http://developers.google.com/image-search>, Retrieved:05.04.2016
10. Imgur API, <https://api.imgur.com/>, Retrieved: 05.04.2016.
11. aXe: the Accessibility Engine, <http://www.deque.com/products/axe/>, Retrieved: 05.04.2016.
12. Techniques for WCAG 2.0, G1: Adding a link at the top of each page that goes directly to the main content area, <https://www.w3.org/TR/WCAG20-TECHS/G1>, Retrieved: 07.04.2016.