

# AUTOMATIC CONVERSION TOOL FOR ACCESSIBLE WEB

Juan Manuel Fernández, Vicenç Soler

*Dept. Microelectrònica i Sistemes Electrònics, Universitat Autònoma de Barcelona, Cerdanyola del Vallès, Spain*

Jordi Roig

*Dept. Microelectrònica i Sistemes Electrònics, Universitat Autònoma de Barcelona, Cerdanyola del Vallès, Spain*

**Keywords:** Web Accessibility, WAI, disabled people, HTML, XHTML, CSS.

**Abstract:** A tool for automatic conversion for making accessible Web pages is presented. It is set in the Help to the Integration of Disabled Persons to the New Technologies field. The project tries to bring the Web to as much people as possible. To get it, we analyze HTML documents and style sheets to adequate them to the regulation of the accessibility dictated by World Wide Web Consortium. The application of this project means a substantial improvement to the access of the information contained in the Web, even though a disability is suffered or not. It improves navigation with all kinds of devices and the view of the Web contents.

## 1 INTRODUCTION

It is undeniable that the importance of New Technologies and its incidence in daily life grow, and Web formats are some of the most important technologies. There is a lack of sensibility towards the different communities of disabled people, and this is the reason to the limited access we can find to these technologies and to the information that the Web contains (WebAim, 2006)(Sidar,2006).

To solve the low Web accessibility there are two possible ways. The first one is to develop some tools that can understand Web contents, which are not accessible to all people, and to show them as accessible information. The second one is to develop accessible contents directly. For this purpose, WAI (World Accessibility Initiative) Guidelines, as a part of the World Wide Web Consortium (W3C), are a set of rules to be followed to ease the access to Internet to people with disabilities (Web Content Guidelines, 2006).

Nowadays, there is a lack of knowledge of these guidelines among the most of the developers. It is the reason why they do not work in an accessible manner, and therefore, to create a tool to transform incorrect content into content which follows the WAI Guidelines is a need.

The application presented in this paper and named Automatic Conversion Tool for Accessible

Web (ACTAW) is set in the Integration of Disabled Persons to the New Technologies field. The application works on Web documents to adapt them to the WAI Guidelines. So, the main goal of this application is to transform the web pages into accessible to be displayed in an existing navigator.

The aim of this project is to eliminate the maximum of the barriers which exist all over the Web. The tool applies the most of the WAI Guidelines to get the maximum of accessibility. The improved accessibility will not only take effect on people with special needs, but everybody can take benefit from these guidelines. The audio or visual disabilities are the most known, but they are not the only ones.

The currently project is the evolution of the “Web Converter for visual disabled people” that was developed in 2005 at the Autonomous University of Barcelona (Soler, 2005). Following it, we started the project with the solutions proposed by this tool; we created new solutions and improved the internal structure of the system. This is the hardest part of development, and also the most important improvement.

The tool offers a system that can analyze a long number of Web pages. Having in mind the lack of correction of the structures of a great amount of Web pages, a good solution is working with HTML tags in a low level, without taking into account its

grammar. This way of working allows us to improve the accessibility of a very great set of Web pages.

This paper is structured in several sections. In the first one, the characteristics of this application will be explained, following by the analyzer, which is the core of this application. Then, the standardisation process is exposed by the solutions applied to every guideline. Finally, the obtained results and conclusions are explained.

## 2 CHARACTERISTICS OF THE APPLICATION

In this section, the most important characteristics of this application will be exposed.

Java is the language used to develop ACTAW, because it allows to create a tool that can be executed in different platforms. The result is a tool totally independent, which makes possible free election of technology by the user.

On the other hand, the independence of the original document's format is really important and complex to obtain. The analysis of any kind of document, independently from its language (HTML in this case), requires a parser with the knowledge of the basic structure and the different elements of the language. This information is in the grammar of the specific language, which in our case is dictated by the W3C.

The problem appears when the existing navigators do not follow the W3C rules and, in addition, they allow the developers write the HTML code with incorrectness, which are interpreted of different ways by every navigator. This brings us to a situation where exist great differences in the structure among HTML documents. If we want to analyze all the possible documents, we have to create a parser that ignores this point, and that works without the HTML grammar, just with HTML tags.

Taking this facts into account, ACTAW stores the information contained in the HTML document in two ways. The first one is to store all information about the known tags and its attributes in an external file. This file can be easily modified and extend, and it is written in XML format. The second one consists on store the tags and/or attributes found in wrong places or unknown, because it is impossible to control all the existing tags and attributes. These new elements are stored like unknown. This solution allows us to save the whole information, even if this is incorrect and, thus, to maintain all the original information.

## 3 THE ANALIZER

The application transforms directly Web documents, developed in HTML, XHTML and CSS languages, to documents in HTML and CSS that follow the most WAI Guidelines as possible. The result documents are the most similar to the original as it is possible.

We can assure that the resultant Web document offers the maximum information about the original Web page, and also the replacement of the incorrect structures by those recommended by W3C. These structures do the same actions and are totally accessible. After these modifications are applied, the document can be visualized in the correct way whether we use a traditional navigator or an adapted navigator to disabled people.

To offer the maximum information, it is necessary to be totally independent of the correctness of the document over our work. For this reason, the tool analyses the entire HTML document and its structure, but it does not follow the W3C HTML's grammar strictly. With that, we can analyse the most amount of Web pages as possible without handicaps like deprecated or unaccepted tags.

The CSS documents are analysed as well, if found, in the same way as the HTML documents. It means that the tool tries to be the most independent of the document's code format.

Before the tool applies the different guidelines, if it is necessary, we modify some aspects to get a uniform document's format. The modifications consist on replacing colours not defined in hexadecimal format, or unaccepted tags in the W3C grammar for its correct equivalent. At last, over the modified document, we apply the WAI Guidelines. This application adds all the necessary information to the different documents, and creates new CSS documents if needed.

## 4 THE PROCES OF STANDARIZATION

Now, we are going to comment the guidelines to create an accessible Web site. As a whole there are 14 WAI guidelines with different techniques to follow. We are going to comment some of them shortly because explains all can be extremely long, only the most interesting solutions will be explained. At the beginning of every guideline its text is presented and then it is explained the implemented solution.

#### **4.1 WAI Guideline 1. Provide Equivalent Alternatives to Auditory and Visual Content**

Provide content that, when presented to the user, conveys essentially the same function or purpose as auditory or visual content.

The susceptible elements to this guideline are the images, image maps and embedded objects. We provide alternative texts with all the information that can be obtained from the HTML code. In these alternative texts, we indicate if these elements are used like links or, what kind of element they are. This information allows the final user to differentiate, for example, the images that are used as links than the ones which have other functions. It also allows us to distinguish among different kinds of embedded elements.

Moreover, we provide alternative texts to pages organized through frames. In these texts we indicate the amount of rows and columns as well as the position they have. Furthermore, we add information to navigators which cannot support navigation through frames.

#### **4.2 WAI Guideline 2. Do not Rely on Colour Alone**

Ensure that text and graphics are understandable when viewed without colour.

We check the defined colours in the style sheets associated to the Web page. If these colours do not surpass the limit marked by the W3C to the contrast colours, the application will create a new CSS document with colours that contrast correctly. This allows us to get the maximum similarity among the original and the resultant document.

#### **4.3 WAI Guideline 3. Use Mark-up and Style Sheets and do so Properly**

Mark up documents with the proper structural elements. Control presentation with style sheets rather than with presentation elements and attributes.

The grammar used in a document is marked with the DOCTYPE tag. The fact that we include this tag, or modify it, is very complex and we have not got it. The tag must be correct from the origin. On the other hand, the tags that mark visually effects, like bold type, are substituted by their equivalents that mark structural effects, like emphasis type. We add contextual clues; which allow the user to follow correctly a list. The list can be nested or simple.

#### **4.4 WAI Guideline 5. Create Tables that Transform Gracefully**

Ensure that tables have necessary mark-up to be transformed by accessible browsers and other user agents.

At this point, we offer to the final user all the possible information about the tables which are contained in the document. This information is the size of the table inside the window and the amount of columns and rows. Altogether this information we group the columns according to its headers, if they are available. The application does not make actions to get the correct alignment of the table. This is because we cannot obtain information about the contents of the table. If we make an incorrect alignment the final user can be confused.

#### **4.5 WAI Guideline 7. Ensure user Control of Time-Sensitive Content Changes**

Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or stopped.

A lot of time-sensitive contents are coded in JavaScript. This kind of code is out of the application's scope.

The animated gif and embedded objects are not possible to stop. This content must be created with care to accomplish the guidelines.

On the other hand, the application substitutes the movement and blinks of a text, created with HTML tags, by CSS techniques and equivalent tags. These tags can be disabled from the navigator too. We delete all the automatic redirections and refreshes as well. This fact avoids that the finally user loses the control of the visited Web pages and their content.

#### **4.6 WAI Guideline 10. Use Interim Solutions**

Use interim accessibility solutions so that assistive technologies and older browsers will operate correctly.

Opening a new window with a link can confuse the user. However, to erase these links can be the cause of losing information. For example, if we want to come back to a previous page and it depends on a user's session, it is possible that this page cannot be visited because the session has been expired. This situation has made us to take the decision to inform the user whenever a new window is opened.

At last, we fill the edit box and the text areas with the texts "Write here" (if it has 12 characters or

more than that) and “Text” (if it has between 12 and 5 characters). If the size is less than 3 characters, we do not write any text because it is impossible to find a clear text.

#### 4.7 WAI Guideline 13. Provide Clear Navigation Mechanisms

Provide clear and consistent navigation mechanisms -- orientation information, navigation bars, a site map, etc. -- to increase the likelihood that a person will find what they are looking for at a site.

The basic mechanism in the Web navigation is the element named link. This must be as clear as possible, and to get this, we add alternative texts with the title and the target URI. In the situation of a section’s link, e-mail’s links and broken links, we indicate it. In addition, we indicate the frame, or window, target. This method provides all the possible information to the user in a clear manner.

On the other hand, an untitled document cannot indicate what content it has and it can disorientate the user. In this situation, we search a header to add it and, if we cannot find any, we add the text “Untitled Document”.

## 5 TESTS AND RESULTS

We have developed a test bench composed by 55 Web pages. We wanted different Web pages and 2 characteristics: the Web page must have an interesting visual design and it must have elements that can make the Web site inaccessible.

The tool analyzes the 100% of the Web pages in the test bench. Moreover, the Web pages improve its accessibility. The 74.54% of the elements are considered acceptable, i.e., either the Web page result is similar visually or the solutions improve the accessibility. Only the 7.27% Web sites of the test bench were accessible, and we improved a 67.27% from this group of Web sites.

So the results have been acceptable in terms of the accessible Web page is similar visually. Anyhow, it is difficult to reach the complete accessibility in a Web page, if it was not thought to be accessible.

## 6 CONCLUSIONS

In this paper, a tool for converting normal web pages to accessible by applying the WAI guidelines has been presented. Another tools are developing now to clean and repair HTML code, like TIDY (TIDY

Project, 2006), and others to complain with standard W3C (Chen, 2006). But the main difference is that this application converts automatically a Web page to accessible.

As a first conclusion, some guidelines were not implemented due to their dependence from the developers. That is, the analyzer has to know the meaning of the information offered with text and images, because to apply the rules when the Web page is created is really hard to do. To solve this problem, a semantic analyzer of the document content is needed.

Even though, considering these situations, we can conclude that the application has attained the initial aim, which was to improve accessibility in all technologies based in HTML, XHTML and CSS. The application is able to offer all the information which is in the code of the document improving navigation and orientation in a Web site.

Nowadays we are working in the expansion of the action field, which consists on developing a JavaScript’s code parser. This parser allows the application to work with a major amount of guidelines and improves accessibility in a Web page that uses this kind of script.

## REFERENCES

- Soler, V., Merino, A., Roig J., 2005. Web Converter for Visual Accessibility. In CPSN’05 - The 2005 International Conference on Computers for People with Special Needs. Proceedings
- Web Content Guidelines Working Group. Web Content Accessibility Guidelines 1.0. Retrieved October 25, 2006, from <http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/>
- WebAIM: Web Accessibility in Mind. Retrieved October 20, 2006 from <http://www.webaim.org>
- Fundación y Seminario Iberoamericano sobre Discapacidad y Accesibilidad en la Red (SIDAR). Retrieved October 15, 2006 from <http://www.sidar.org>
- TIDY Project. Retrieved October 2006. <http://tidy.sourceforge.net>
- Kirchner, M. Evaluation, Repair, and Transformation of Web Pages for Web Content Accessibility. Review of Some Available Tools. Proceedings of the Fourth International Workshop on Web Site Evolution (WSE’02), pp.65, 2002.
- Chen B., V.Y.Shen. Transforming Web Pages to Become Standard-Compliant through Reverse Engineering. Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility?, pp. 14-22. 2006.