Guiding accessibility issues in the design of Web 2.0 sites

Lourdes Moreno
Computer Science Department
Universidad Carlos III de Madrid
Av. Universidad 30 - 28911
Madrid, Spain
lmoreno@inf.uc3m.es

Paloma Martínez
Computer Science Department
Universidad Carlos III de Madrid
Av. Universidad 30 - 28911
Madrid, Spain
pmf@inf.uc3m.es

Belén Ruiz
Computer Science Department
Universidad Carlos III de Madrid
Av. Universidad 30 - 28911
Madrid, Spain
bruiz@inf.uc3m.es

ABSTRACT

To include accessibility in the design of Web 1.0 sites entails difficulties, but this situation becomes more complicated when the user become creator in the web. There are new requirement to be considered on the design. Due to user’s active interaction in web 2.0 environment, where they are not only information receptors, but issuers as well, it is necessary to articulate mechanisms of accessibility. This paper offers a solution to include accessibility requirements in the design process based on using accessible templates for contents to preserve accessibility and the editing process using models and rules to annotate structure and semantics of contents. This annotation will be added to the final codification, which will guarantee a better compliment with the standards and accessibility guidelines.

Categories and Subject Descriptors

General Terms
Design, Human Factors.

Keywords
Web accessibility, Web engineering, requirement, editing, design, blog, template.

1. INTRODUCTION

The roles of the different actors playing in the World Wide Web are changing; the triumph and increment of the web sites 2.0 such as blogs bring us to reflect on whether these changes are a problem or not to achieve accessible web sites and as a consequence of this if this situation is going to mean the exclusion of some user groups.

Focusing on what happens when surfers become authors and designers, the main change is that the user has gone from being a passive reader to play a main role as a creator of contents. This way, the user takes the role of editor, but also of contents designer. This new perspective poses new problems concerning accessibility:

- The user may not have training on standards and accessibility, and consequently this produce undesirable effects on web pages.
- The editing tools generate final code not considering accessibility issues.
- The user may not be able to use the editing tools, i.e. the editing tool is not accessible in its use.

From this point of view we offer a solution based on guiding contents design using accessible templates to preserve accessibility. The user will be limited to how he can edit his contents but he will be offered at the same time the control over what he designs and edits in each moment. To achieve this, the system must include assistant tools which have to be integrated into the editing process and which will guide the content editors.

2. RELATED WORK

The web accessibility initiative (WAI) [9] and its standards such as Authoring Tool Accessibility Guidelines (ATAG), User Agent Accessibility Guidelines (UAAG) and Web Content Accessibility Guidelines (WCAG) [10] are considered, especially in the case of WCAG as rules where the legislation and politics can directly or indirectly refer to. Although they don’t cover all the situations and the resources may be inaccessible even though they adjust to the directives [1], they will be the standards. The benefits of WCAG guidelines are well known, and their help is undoubtedly fundamental for designers, evaluators, etc.

In the area of authoring tools in the web where accessibility problems predominate some aspects such as the compatibility of the web editors with platforms and user agents have to be considered. However there are solutions arising which can offer opportunities towards the accessibility in the new interactive paradigms [7]. Content Management System (CMS) integrate two approaches, WYSIWYG (What You See Is What You Get) and WYSIWYM (What You See Is What You Mean). The editor WYSIWYG is exclusively based on presentation, and the WYSIWYM is the most recommendable format as it facilitates a necessary semantic structure to offer accessibility. With a similar approach some works propose structural templates to be integrated in authoring tools [2].

From an engineering point of view the development process of accessible web applications or sites does not require a specific...
methodology in itself; it only makes sense when the methodological approach includes accessibility criteria. There are proposals focused on web engineering (OOHDM, UWE, OO-H, OOWS and WebML amongst others.) with processes, models and adequate techniques. There are approximations on WCAG have been found as a work frame for the development process in the whole life cycle [3].

Related to the WCAG formalization as is proposed in this paper, there are some works such as applying semantic web technologies [5], and the research based on XLS Transformations [4].

This work is part of a research study, Accessibility for Web Applications (AWA). Its objective is to offer a methodological support to integrate accessibility from a Web engineering perspective in the whole life cycle of a web application. In this case, we are dealing with the accessibility in the editing process by users. The AWA focus using abstraction helps to detect all the affected components in the development processes, and to enable mechanisms of accessibility. It has a modular structure to facilitate the extension, distinguishing as main components which can be affected by accessibility: human, technological, normative and web object components.

Our approach is basically based on the fundamental aspect of considering the accessibility in the analysis phase as a functional requirement and on articulate techniques following the methodologies. A requirement in AWA is that mechanisms must be defined and implemented to preserve the accessibility.

3. **AWA APPROACH: ACCESSIBILITY COMPONENTS AND MECHANISMS IN THE WEB EDITING PROCESS**

In this scenario of editing process, it is assumed that it is supported by a system which doesn’t go against the accessibility when choosing technologies (based on standards), architecture of the application (n-layer models separating the business logic from the presentation), and trying to avoid conflicts between authoring tools and user agents.

The analysis phase of a project that defines the functional requirements has to guarantee preservation of accessibility in the editing processes of web contents by users.

The initial solution proposed by the analysts is shown below in natural language and by a UML use case diagram (Figure 1).

**Follow a design and development guided by templates. These templates should be accessible according to WCAG.** The templates should be flexible allowing the editing of contents which adjust to a logical structure and should avoid accessibility problems. To achieve this, the templates have to organize elements in nested boxes so that in each box of an element we have the boxes of the elements which form it and this behaviour repeats recursively itself.

In the editing process, the user must be given the option to select a template to edit his contents in a box, by means of an accessible editing tool. The editing tool should guide and assist the user with the aim to comply the WCAG. Finally there will be a transformation of the edited content to a final accessible code.

Following the previous solution and to provide traceability in compliance with functional requirements to be included in the development process, we have studied what technological and human components are affected by the accessibility, in order to enable the necessary **Accessibility Mechanisms**.

**Figure 1: Use case of editing process on the web**

### 3.1 Editor component and WCAG model mechanism.

WYSIWYG editors predominate in the creation and editing of web contents. They are included in CMSs and the web sites 2.0 where the user edits his own contents. This type of editors allows the user to carry out a great number of operations which constitutes a problem for accessibility due to the fact that the final resulting code will depend on the user training as well as on the code that the editor generates. As an alternative to WYSIWYG, there also exits WYSIWYM approaches guided by the semantic structure instead of by its presentation. There are various elements with this semantic information in HTML documents such as language, headings, abbreviations, etc. These elements are semantic information labelled and are necessary to comply with the WCAG (2.4.10, 3.1.10, 3.1.4 success criteria of WCAG 2.0), and its non-compliance, for instance in the case of the headings affect accessibility due to the fact that someone who navigates by headings will find himself confused[8].

These accessibility problems have been incremented by the use of WYSIWYG in the creation of web pages because users without training in accessibility can use semantic elements to page design. With WYSIWYM editors these problems can be reduced. To include content to a web page using WYSIWYM editors, there is a pre-requirement of knowing how the pages are structured. For example, the web pages of a blog are structured (see figure 2) with a title, author, and summary and may have an image with a foot image, etc. The templates must be defined on the web site
where each editing box is associated with a logical structural model.

Although the two types of editors could be used, the WYSIWYM editor is more appropriate to control accessibility criteria. On the other hand, in the case of blogs, wiki’s, social web, etc. the user is who edits and designs, and their main preoccupation is the presentation. So, in this proposal, the editor should be oriented to the user as a WYSIWYG type, but to enable mechanisms as WYSIWYM editor to obtain an accessible and valid code.

The editor is a Component affected by Accessibility. It should be accessible and can be used by all user profiles. The inclusion of this type of web sites should be assured; the Accessibility Mechanisms must enable the implementation of accessible editing tools in their use, considering access by Assistive Technologies for people with disabilities.

The editor must integrate an assistant into the editing process to guide the user. It must be designed having the user in mind with usability criteria. A modules that can be included in the system to help the user is a dictionary module for automatically extend the abbreviation (according 3.1.4 success criteria of WCAG 2.0). It’s important to satisfy and give a sense of control to the user, it is not recommended to give too many options of editing to not overwhelm the user and to show the no active editing options.

To limit the action of the user when interacts with the editor it is convenient to enable Accessibility Mechanism not offering determined operations which can create accessibility problems, such as the editing of tables, forms, etc. not accessible; or enabling the Accessibility Mechanism which permits these operations, but controlling all the necessary factors to comply with the WCAG. It is the case of editing an image where the user must request that the user introduces an alternative text according to Guideline 1.1 of WCAG 2.0, or if a video is permitted, then the editor should force the user to facilitate an XML file with the alternative content of the caption. Although in most cases, it is not an automatic annotation and depends on the user intervention.

For these Accessibility Mechanisms, models and rules according to comply with WCAG in a higher grade must be defined, and as a main objective: “it shouldn’t permit that the editing tool generates a code that gives problems of accessibility during the process, it must generate a valid code”. For instance, a image element, a set of properties or items must be defined in order to comply with guideline 1 of WCAG. The Figure 4 shows an approximation to the image element and the properties that must be encapsulated in order to get the accessibility used XML [6].

```xml
<!ENTITY logo SYSTEM "url" NDATA JPEG >
<!ENTITY sound SYSTEM "url" NDATA MP3 >
<ELEMENT audioimage (image, alt, longdesc)
  <ELEMENT image (figure, audio)>
  <ELEMENT figure logo>
  <ELEMENT audio sound>
  <ATTLIST   logo text CDATA #REQUIRED
              lang xml:lang #REQUIRED>
  <ELEMENT longdesc #PCDATA>
  <ATTLIST   longdesc text CDATA #REQUIRED
              lang xml:lang #REQUIRED>
  <ELEMENT audio>
  <ELEMENT image PCDATA>
```

Figure 4: accessibility semantic metadata for image element

If a image is to be included in a web page, the WCAG properties that must be required are “alt” and “longdesc” (figure 4). The “alt” property contains a string, the “longdesc” property provides a description, and all of them must be declared as mandatory in order to carry out the first level of conformance of the guideline 1 of WCAG. Also, the language for the content of these properties is introduced by the data type “xml:lang”. Additionally, the image element has associated an audio element for adding description audio and sound to it. If this scheme is created and all images in the web fit to it, we ensure the accessibility in the every image.

3.2 Template component and semantic structure model mechanism.

The user will edit contents on a box of available templates in a web site. The template is another technological Component affected by Accessibility and it must enable an Accessibility Mechanism so that these templates are designed according to WCAG. These templates in XHTML should have the content completely separated from the presentation with accessible Cascading Style Sheets (CSS). These templates should also give support to the editing tool and should follow some models previously created, which define the logical semantic structure of each one of the boxes in relation to the contents that it can store. The editing tool should only permit to edit contents that can adjust to these models.
3.3 Web page component and transformation mechanism.
When editing contents, a process of semantic labelling is taken place on them. The enriched contents resulting with metadata accessibility could be stored as XML format with a XML Schema definition. The enriched contents will contain various elements with its structural semantic annotation following the models, and each one will have some rules on how to transform the final resulting code. These rules and constraints are Accessibility Mechanisms due to the fact that it assures a transformation of these contents to a final accessible code.

Applying these transformation rules can be technologically solved with XSL Transformations (XSLT), which transforms each XML element in its final accessible component XHTML code according to the WCAG. This last step does not require human intervention or manual annotations.

Figure 3 shows the editing process, with the Components affected by Accessibility and Accessibility Mechanism.

4. ACCESSIBILITY REQUIREMENTS IN THE DEVELOPMENT PROCESS
A development iterative process has been defined, with the analysis, design, development, launch and maintenance phases of the whole life cycle of the application. In this work we have exposed a vision as far as accessibility is concerned; each project has its requirements and development methodologies, the accessibility requirements and its consecution will have to be integrated from the beginning of the development process.

In the analysis phase, where the functional requirements have been defined in form of Components and Accessibility Mechanisms, each one of them has points of execution in the different phases of the process (see Table 1). The components and mechanisms are inter-dependent among them, and there are components directly affected in one phase, and indirectly in the other ones.

Apart from the artefacts, there are different roles (analyst, designer, evaluator, user...) to consider, and have been detected responsibilities and tasks for them as far as accessibility is concerned.

4.1 Evaluation of the accessibility guided by templates
A continuous evaluation must be carried out in each one of the artefacts of the development process. The added value of this proposal is to reduce efforts in the evaluation process, because it is not necessary to review all WCAG checkpoints in the Web pages. In the design guided by accessible templates, the editing of contents is converted in a controlled process. These systems can reduce the effort of evaluation in accessibility issues. Besides is a solution in the domain of wiki, blogs, social networks, etc. where the user is who edits and there are no modifications on the content edited, although there are always exceptions and depends on human factors.

5. CONCLUSIONS
With this new situation where there are new forms to create contents where the user can design and edit own contents technological solutions must be given so not to leave the accessibility in hands of the user nor in those of the editing tool of the contents. A solution has been presented based on the design guided by templates with the use of models, to obtain a semantic structure and WCAG annotation in the contents, necessary to elaborate web pages to comply the WCAG.
### Table 1. - Accessibility Mechanisms in the process.

<table>
<thead>
<tr>
<th>AWA-Component</th>
<th>AWA-Accessibility Mechanisms in the analysis phase</th>
<th>AWA-Mechanisms in the design phase</th>
<th>AWA-Mechanisms in the implementation phase</th>
<th>Direct Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1.- Editor</td>
<td>M1.1. - WYSIWYG Editor, but offering a semantic structure and accessibility criteria in an integrated tool assistant</td>
<td>- Design of the Editing tool (with an assistant functionality)</td>
<td>- Development of the editor tool (with an assistant functionality)</td>
<td>M2.1. M1.3</td>
</tr>
<tr>
<td></td>
<td>M1.2. - Universal design, usability and accessibility criteria.</td>
<td>- Design of the editor tool (of accessible use)</td>
<td>- Development of the editor tool (of accessible use)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M1.3. - WCAG Models</td>
<td>- Design of the model</td>
<td>- Development of models, rules and constraints.</td>
<td>M3.1 M3.2</td>
</tr>
<tr>
<td>C2.- Template</td>
<td>M2.1. - Accessible design (nested boxes)</td>
<td>- Design and implementation in XHTML and CSS to follow standards and WCAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M2.2. - Structural semantic Models</td>
<td>- Design and implementation of the model, rules and constraints</td>
<td>M1.1 M1.2 M1.3 M3.1.</td>
<td></td>
</tr>
<tr>
<td>C3.- Final Web page</td>
<td>M3.1. - Formalization for web content with the annotation.</td>
<td>- Metadata definition in XML and the XML_Schema</td>
<td></td>
<td>M3.1 M2.2</td>
</tr>
<tr>
<td></td>
<td>M3.2. - Transformation to a final valid and accessible code according to WCAG</td>
<td>- Design and implementation of the rules and constraints (XSLT Transformation)</td>
<td></td>
<td>M3.1 M1.3</td>
</tr>
</tbody>
</table>

### 6. FUTURE LINES

This work is based on the AWA investigation, which has as its objective the increment in the level of abstraction in the components and methods to follow to achieve accessibility. The main target is to offer guidelines and methodological supports from an engineering perspective.

To reduce on costs and to offer flexibility in the development, a possibility to look into is to use guided developments for models with Model-Driven Architecture (MDA), as these models and rules which give support to the WCAG model as the models with knowledge from the logical structure semantics of the web pages, although they are dependent on this latter case of the web application specifies, it can be used in different domains.

### 7. ACKNOWLEDGMENTS

This work has been partially supported by the Regional Government of Madrid under the Research Network MAVIR (S-0505/TIC-0267) and by the TIN2007-67407-C03-01 project (BRAVO: Advanced Multimodal and Multilingual Question Answering).

### 8. REFERENCES


http://doi.acm.org/10.1145/1243441.1243473
