Methodology for identifying and solving accessibility related issues in Web Content Management System environments

Juan Miguel López
Department of Computer Languages and Systems,
School of Engineering,
University of the Basque Country (Spain)
juanmiguel.lopez@ehu.es

Afra Pascual[1], Cristina Menduiña[2], Toni Granollers[3]
GRIHO research group,
Polytechnic School,
University of Lleida (Spain)
apascual@diei.udl.cat[1]
cmenduinya@alumnes.udl.cat[2]
tonig@diei.udl.cat[3]

ABSTRACT
This work presents a methodology that allows identifying and solving accessibility related issues in web pages using Web Content Management System (CMS) environments. In this sense, the methodology establishes a series of steps to be performed in order to ensure that the content managed by CMSs is accessible. A study has been performed on two different CMSs to check the validity of the steps defined in the methodology. The paper includes the methodology used, the evaluation performed on both CMS (OpenCMS and Typo3) and the key findings of the analysis. The results of the study have been positive as the objective of providing CMS environments that allow developing accessible web pages has been fulfilled.

Categories and Subject Descriptors
H.1.2 User/Machine Systems: Human factors & human information processing
K.5 LEGAL ASPECTS OF COMPUTING

General Terms
Management, Measurement, Experimentation

Keywords
Web Science, Web Engineering, Web Accessibility, Web Accessibility Evaluation

1. INTRODUCTION
Legislation about the fulfilment of accessibility requirements is currently in force in many countries around the world. For instance, Section 508 [1] requires federal agencies in the United States to make their electronic and information technology accessible for people with disabilities. These kinds of legislative changes have also been made in more countries around the world, such as European Union countries [2]. These legal frameworks not only affect public administrations, they also affect corporations that develop software or provide services for their use in civil service.

W4A2012 Technical Paper, April 16-17, 2012.Lyon, France.

.

In this context, it must be taken into account that there are a growing number of institutions that use Web Content Management Systems (CMS henceforth) to manage the Web content on their websites. These kinds of systems are especially useful as they can be used by users with no Web programming skills to introduce or modify Web content in the website. This approach has some risks regarding accessibility. On the one hand, CMS themselves may not provide appropriate support for developing accessible websites. On the other hand, including or modifying Web content by unskilled users may lead to a situation in which accessible websites can turn into non accessible.

Given the importance that CMS systems are gaining, this work presents a methodology that allows identifying and solving accessibility related issues in web pages using CMS environments. The methodology is focused on which steps are necessary for developing accessible web pages in CMS environments. A study has been performed on two different CMS to check the correctness of the steps defined in the methodology. By following the steps defined in the methodology, both CMS environments have been configured taking accessibility related issues into account. As a result of this process, both CMS have provided a proper environment for users to manage accessible web pages.

The rest of the paper is structured as follows. Section 2 describes the background for this work. The methodology to provide proper CMS environments for managing accessible web pages is introduced next. In Section 4 the insights of the evaluation performed on two CMS environments are explained. Then, a discussion about the performed study is set out. Finally, the conclusions of the paper are shown.

2. RELATED WORK
An accessibility evaluation of web pages has been traditionally performed revising the fulfilment of Web Content Accessibility Guidelines (WCAG henceforth) [3][4] proposed by the World Wide Web Consortium. In order to fulfil an accessibility evaluation for a website, (X) HTML content and CSS style sheet linked to each web page are reviewed with automatic accessibility evaluation tools [5], which show a revision of automatically detected problems and manual revision by accessibility evaluators. On the other hand, authoring tools are software and services that people use to produce Web pages and Web content, for instance CMS. The Authoring Tool Accessibility Guidelines (ATAG henceforth) define how authoring tools should help Web

developers produce Web content that is accessible and conforms to WCAG [6] [7].

In this sense, analyzing user interfaces to manage Web content provided by CMSs is critical for a proper accessibility evaluation [8]. The CMS transfers the content into the templates, processes various sites and links these automatically to a consistent website [9]. However, several issues have been detected in CMS concerning accessibility [10] [11]. Information providers, in this case the CMS users, are not usually aware of the need for creating accessible content, and the tools do not provide information to support users in this feature. Furthermore, CMS do not usually enforce users in aspects that should be taken into account in order to generate accessible Web content [12]. Related with these aspects, there are important threats regarding accessibility specific to CMS environments. In this sense, errors may be systemic to all the web pages managed by the CMS and it can be very costly to solve these issues when the amount of web pages is as big as the one that can be managed by a CMS. There are some works related with methodological solutions for ensuring accessibility in Web environments and preventive maintenance [13], [14] and [15]. In this context, [16] proposes a framework that provides information about possible corrective accessibility maintenance activities for webmasters based on logs from the CMS environment. Some online articles, such as [17] and [18], show information about improving accessibility in CMS environments, but they focus on specific solutions for specific cases and they lack a proper methodologically that is sound in evaluation procedures that could be extended to other CMS environments.

3. METHODOLOGY

With the aim of supporting accessibility of any kind of interactive system, it must be taken into account in all the stages of the software to be developed. Several existing methodological approaches include accessibility as a main factor to be taken into account from the beginning of a software lifecycle to the end [13].

3.1 Methodology description

Proposed methodology is based on which steps are necessary to perform in order to provide a proper CMS environment for managing accessible web pages. The methodology is iterative, as steps back can be performed when accessibility objectives are not fulfilled or requirements in the CMS environment changes. The methodology employs the accessibility evaluation methodology defined by the W3C [19], Web content accessibility guidelines (WCAG) [3] [4], accessibility guidelines for authoring tools (ATAG) [6] [7] and takes into account the internal functioning and structure of CMS environments. The proposed methodology includes using automated accessibility evaluation tools as part of the process. The different steps proposed in the methodology are displayed in Figure 1 and described next.

3.1.1 Step 1: Select CMS and configuration

Choosing one CMS and its configuration depend on several factors, and may not always be a decision that depends on the CMS administrators. It is always necessary to configure the environment with all possible accessibility characteristics included. If CMS selection depends on its administrators, they should take into account their expertise on different CMS environments and the accessibility support every CMS has.

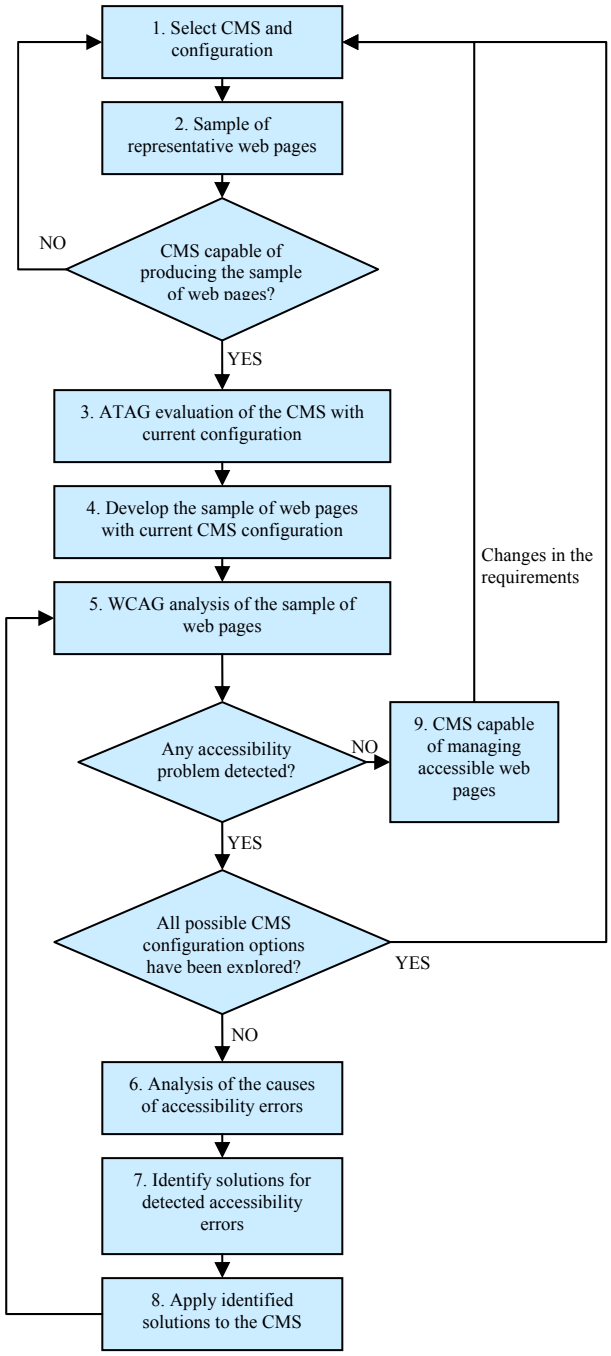

Figure 1. Steps in the methodology

By following the methodology, CMS administrators can return to step 1 after at least one iteration in the process has been fulfilled if selected CMS configuration has no chance to provide proper accessibility for the given sample of web pages (decision to be made after step 2). Furthermore, it could imply changing selected CMS if the administrators finds no way to configure it properly with the aim of providing accessible web content (decision to be made after step 5). Another chance to return to this step would be that requirements regarding the type of Web content users can manage change and, therefore, configuration changes or changing the CMS in order to meet the new requirements are required (decision to be made after step 9).

3.1.2 Step 2: Sample of representative web pages

A sample of representative web pages to be developed by people using the CMS must be planned in this step. It is important that all possible HTML tags and different layout combinations for the web pages that would later be developed using the CMS are present and identified in the sample. This aspect is important for ensuring that an accessibility evaluation of the sample would be able to identify all possible accessibility errors in every web page managed by the CMS [19].

Regarding the type of Web content intended to be developed by the users of the CMS, the basic CMS installation may not be enough. It must be noted that CMS must be able to meet the requirements for the CMS to be able to manage all the kinds of web pages users should be able to create and edit by using the CMS environment. If the content includes elements that cannot be managed by the basic configuration of the CMS, configuration modifications should be made and plugins or modules should be added for allowing the CMS to manage the type of Web content it is intended to manage. If a proper environment for meeting such requirements cannot be provided by current CMS, a jump back to step 1 would be necessary. Otherwise, we should proceed to step 3.

3.1.3 Step 3: ATAG evaluation of the CMS

As CMSs are authoring tools that allow managing Web content, it is necessary to evaluate them by using ATAG guidelines. Although the evaluation of ATAG is not compulsory for ensuring that an authoring tool produces accessible Web content, it provides a valuable source of information for identifying possible Web accessibility related problems within the CMS [20]. Furthermore, ATAG evaluation provides information that can be used as a source of feedback for CMS developers, so they can improve their support for accessibility.

It must be noted that ATAG validates authoring tools to use for people with disabilities, but the main target of the methodology is to ensure that the CMS provides a proper environment to generate accessible content in Web pages. In this sense, it is possible that not all ATAG guidelines could be solved and that the Web content pages are accessible [8]. As a result, CMS may be valid for providing accessible web pages, but not for being managed by people with disabilities.

Besides, in some cases it would not be possible for CMS administrators to provide a fully ATAG-compliant accessible CMS. In this sense, CMSs themselves and their related components may not have support for fulfilling all ATAG checkpoints. Furthermore, providing such compliance could be out of the scope of website administrators. As it would imply modifying the CMS itself or its components, it would be the work of CMS developers to provide such support. Nevertheless, it is always desirable to explore ways to improve ATAG compliance and an ATAG fulfilling CMS is much more likely to be helpful for achieving the main goal of providing accessible web page management.

3.1.4 Step 4: Develop the sample of web pages

At this stage, the characteristics of the sample of web pages are clear and we already know that the current configuration of the CMS is capable of correctly handling all the content in them (defined in step 2). Therefore, in this step it is necessary to create a sample of web pages that have to be developed by using the CMS with its current configuration.

3.1.5 Step 5: WCAG analysis of the sample of web pages

The accessibility of the sample of web pages developed by using the CMS in the previous step must be evaluated next. Following the W3C accessibility evaluation methodology [19], it is necessary to use at least two evaluation tools [5] for each web page. It helps to catch potential misidentification of accessibility problems that might result from using a single evaluation tool. The evaluation should be performed using WCAG 1.0 [3] or 2.0 [4] guidelines. Choosing a set of guidelines may depend on local regulations. For instance, legal frameworks in Spain [22] and Italy [23] are still based on WCAG 1.0. As a result of using evaluation tools, a list of accessibility errors and potential errors is obtained.

Depending on the possible errors detected, the methodology establishes different options. In case an accessibility error is not detected, we should jump to step 9, as the CMS would have shown it is capable of managing web pages that follow the pattern of the web pages in the sample in an accessible way. If accessibility errors have been detected, we should check whether more configuration changes could be applied to the CMS to improve the accessibility of managed web pages. In case no more changes can be applied (as all possibilities have already been considered), the CMS would have shown it is not capable of managing accessible web pages, so jumping to step 1 in order to consider the use of another CMS is advocated. Otherwise, as more CMS configuration would be possible, the causes of accessibility errors within the CMS environment must be analyzed in step 6.

3.1.6 Step 6: Analysis of the causes of accessibility errors

Based on the list of potential errors for web pages obtained in the previous step, the causes of these errors must be analyzed in the context of the current CMS configuration. In order to determine the causes of each detected error, different sources of information must be used. On the one hand, the results of the ATAG evaluation (step 3) can provide hints on aspects that have an influence on the accessibility of web pages. On the other hand, the internal structure of the CMS and its workflow must be analyzed to check how the web pages are composed and determine which element introduces inaccessible code in the web pages. In this sense, we must study which CMS components have an influence in the code with accessibility errors.

3.1.7 Step 7: Identify solutions for detected accessibility errors

Once the elements for each potential error have been targeted, possible solutions for each case must be analyzed. In this sense, a solution must not focus simply on general causes for the error (for instance, that a concrete error is caused by the template the CMS uses). It should get into detail (for instance, which part of the template creates the problem and in what way can the template be modified to solve the problem).

3.1.8 Step 8: Apply identified solutions to the CMS

All possible solutions for every accessibility error detected in the previous step must be applied in this step. It can imply modifying the configuration of the CMS, by configuring already present CMS elements or including new ones to the CMS. After all CMS modifications have been performed, we must jump back to step 5, to evaluate the accessibility of the sample of web pages using current CMS configuration to determine if they are accessible.

3.1.9 Step 9: CMS capable of managing accessible web pages

Reaching this step in the methodology implies that the CMS can manage accessible web pages that follow the same pattern as the sample of web pages defined. However, it must be taken into account that developing accessible web pages will ultimately depend on the users that employ the CMS [16]. Therefore, it does not necessarily mean that every web page should be accessible, just that the CMS provides a proper environment for users to manage accessible web pages.

This step is intended to be the last one, unless any change is made in the requirements for the accessibility of the web pages managed by the current configuration of the CMS. If any change occurs, we should go to step 1. This way, the methodology will provide support in case the requirements change.

The possible reasons for changes in the requirements may be different. On the one hand, if new elements must be included in the web pages to be managed by users the whole process should be repeated. However, in this case the process should focus on the new elements; if no changes have been included, the configuration should be valid for all previous kinds of elements. On the other hand, new requirements regarding changing current CMS or modifying its configuration can be made by a client or organization. A migration to other CMS or including new plugins in CMS environments can be considered as examples.

4. CASE STUDIO

The objective of the study is two-fold. On the one hand, it performs an evaluation on two CMS environments by following established methodology to obtain two CMS environments capable of managing accessible web pages. On the other hand, by performing the study it is intended to establish the validity of the proposed methodology.

4.1 Material

4.1.1 CMS and configuration

Two different CMSs, OpenCMS (version 7.5.1)[1] and Typo3 (version 4.5.2)[2], have been employed in the case study, both a with default configuration.

An HTML editor was used on the front end of each CMS to create the content page and the templates are resources on the back end of the CMS (like HTML pages) that are used to separate content from design. With OpenCMS, its default HTML editor (FCKEditor[3]) was used. A template named "Template one"[4] was selected among the different templates available on the default configuration of OpenCMS. Regarding Typo3, its default HTML editor (RTE[5]) and the default template were used.

4.1.2 Representative web pages

In regard to the sample of web pages, a set of 10 web pages was created as a proper sample of the web pages of a Medical School (see Figure 2). These web pages were developed so they included all possible web content that users of a Medical School website could include, focusing on the use of different elements on every web page. All the set of web pages included basic HTML elements such as Headers (<h1>) and Paragraphs (<p>). Table 1 displays the list of pages with the different HTML elements they contained.

Figure 2. Sample of web page

Table 1. List of pages with the different HTML elements they contain. All pages have headers and paragraphs

Web Page	Content elements HTML description
All pages	Headers (<h1>) and paragraphs (<p>)
1 - Presentation	No extra elements
2 - Structure of the center	Bold text format ()
3 - Academic authorities	No extra elements
4 - Mural of the study room	Internal links (<a> <a name>), Images (<img>), Ordered lists ( ).
5 – Syllabus	Tables (<table> <caption> <thead> <tbody> <th> <td> <tr>), Different text formats: text centred ( ), bold (), color font (), color background ().
6 – Departments	Externals links (<a> <a target="_blank">), Unordered list ( ).
7 - Official registration	Externals links (<a> <a target="_blank">), Image link (<a><img>).
8 - News	Externals links (<a> <a target="_blank">), Emphasized text format ().
9 - Contact information	Bold text format (), Address (<address>), Email links (<a href="mailto:...">).
10 - Questions & suggestions	Form elements (<label> <input> <select> <option> <textarea>).

4.2 Procedure

We followed the methodology provided in the previous section. In this case, a single iteration was conducted, as solutions to detect accessibility problems provided in step 7 and implemented in step 8 solved all accessibility problems without the need of implementing more steps. The whole process is explained step by step.

[1] OpenCMS(version 7.5.1). http://www.opencms.org
[2] Typo3 (version 4.5.2). http://typo3.org/
[3] FCKEditor for OpenCMS. http://opencms-wiki.org/FCKEditor
[4] Template one for OpenCMS. http://opencms-wiki.org/Using_existing_Template_One
[5] Rich Text Editor (RTE) for Typo3. http://www.kevinroth.com/rte/demo.htm

- **Step 1:** OpenCMS and Typo3 were installed with a default installation.
- **Step 2:** The sample of web pages for the Medical School was selected (Table 1). In this sense, there was no need to return to step 1, as all elements in the sample of web pages could be correctly developed by default configuration with both CMSs without requiring any further modifications
- **Step 3:** ATAG evaluations were performed on both CMSs. Evaluations were performed by using the ATAG 1.0 [6] by evaluating all checkpoints in A and AA levels of compliance. (see section 4.3.1 ATAG evaluation).
- **Step 4:** The web pages in the sample were developed using the default configuration in both CMSs.
- **Step 5:** The set of pages was evaluated using the WCAG 2.0 [4]. The evaluation was carried out by evaluating all checkpoints in A and AA levels of compliance. The set of web pages were analyzed by using two automatic accessibility evaluation tools, AChecker[6] and TAW[7]. Their reports included both accessibility guidelines with errors, including potential ones. As accessibility errors were detected (see section 4.3.2 WCAG evaluation) and configuration modifications were possible for both CMSs, it was determined that the next step would be 6.
- **Step 6:** The causes for the accessibility errors detected were determined in the workflow of the CMS. In this sense, ATAG evaluation results, internal and WCAG evaluation results were used. As a result, four different error sources were detected (user management, template, HTML editor, internal management of the CMS) (see section 4.3.3 Analysis of the causes of accessibility errors).
- **Step 7:** Possible solutions to provide better accessibility were analyzed in both CMSs for each checkpoint that caused errors. The main solutions found are briefly explained in the next subsection (see subsection 4.3.4 Solutions for detected accessibility errors).
- **Step 8:** The solutions were applied by reconfiguring and installing components in both CMSs.
- **Step 5:** The accessibility of the sample of web pages was evaluated again. It was aimed to check whether current CMS configuration would have allowed creating fully accessible web pages. As all web pages proved to be accessible for both CMSs, the next step was 9.
- **Step 9:** Both CMSs provided accessible web page development environments for the kind of web pages in the sample. In this sense, unless more requirements for the web pages developed using both systems are required, the objective of providing a proper environment for managing accessible web pages that follow the patterns established in the sample was fulfilled.

4.3 Results

This subsection provides the results obtained throughout the process carried out. Results are mentioned based on the step they were obtained.

4.3.1 ATAG evaluation

In step 3, an ATAG evaluation was performed on both CMS with their default configuration. Table 2 displays the quantitative results of the ATAG evaluation. Results are shown in terms of the levels of compliance (A and AA). Values are expressed in terms of the percentage of correctly fulfilled checkpoints for every level of compliance. This results shows that default configuration on OpenCMS is more accessible than Typo3 as there are less errors in ATAG evaluation. Table 3 displays the concrete errors found in the ATAG evaluation.

ATAG checkpoints with no errors in both CMSs are listed below:

- Produce accessible content in the markup language and change the presentation within editing views without affecting the document markup (A level: 1.1, 7.2).

- Allow the author to preserve markup not recognized by the CMS and enable editing of the structure of the document in an accessible fashion and to search within editing views (AA level: 4.3, 7.5, 7.6).

ATAG checkpoints with errors in both CMSs are listed below

- It does not preserve all accessibility information during authoring, transformations, and conversions; it does not generate a valid markup automatically; it does not generate equivalent alternatives; it does not have features that promote the production of accessible content; it does not allow editing all properties of each element and object in an accessible fashion (A level: 1.2, 2.2, 3.4, 6.1, 7.3).

- I does not use the latest versions of W3C recommendations about markup language; it does not integrate accessibility features into the overall look and feel of the CMS; there are elements, such javascript and frames, that do not ensure proper validations of the WCAG to generate accessible pages; help documentation does not exist regarding items that fulfil WCAG (AA level: 2.1, 5.1, 5.2, 6.2).

Only OpenCMS does not allow keyboard navigation in the editor view (AA level: 7.4) while no errors were found only in Typo3 that were not present in OpenCMS.

Table 2. Quantitative results of ATAG evaluation

	OpenCMS		Typo3	
	A	AA	A	AA
Percentage of correct checkpoints	25,00 %	42,86 %	37,50 %	42,86 %

Table 3. Concrete checkpoints errors found

	Checkpoints	
Errors	**A**	**AA**
Checkpoints with no error in both CMSs	1.1, 7.2	4.3, 7.5, 7.6
Errors found in both CMS	1.2, 2.2, 3.4, 6.1, 7.3	2.1, 5.1, 5.2, 6.2
Errors only on OpenCMS	No errors	7.4
Errors only on Typo3	No errors	

4.3.2 WCAG evaluation

In step 5, the accessibility evaluation of the 10 web pages in the sample was performed. Table 4 displays the errors and potential errors detected. As the factors that intervene can be different for every CMS, different checkpoints can be potentially erroneous, so different aspects are put in the table for each. Results of WCAG analysis do not show the number of occurrences of each error. In this case, these data have been considered less relevant than the fact that a concrete checkpoint was not fulfilled. It must be taken into account that, as every web page had a different structure and

[6] Web Accessibility Checker. http://achecker.ca/checker/index.php

[7] TAW Evaluator: http://www.tawdis.net/

elements, different accessibility errors can be found for each of them.

Table 4. Checkpoints with errors in the WCAG evaluation for every web page in the sample

	Checkpoints	
Errors	**A**	**AA**
Errors found in both CMS	1.3.1, 2.1.1, 2.4.4, 3.3.2, 4.1.1 User errors: 1.1.1, 1.4.1, 2.4.2, 3.2.3	2.4.5, 3.3.4 (only page 10)
Common errors on OpenCMS web pages	3.1.1, 2.4.1	No errors
Errors on some web pages on OpenCMS	1.3.3, 3.3.1 (only page 10)	1.4.4 (only pages: 2, 5, 8 and 9), 3.3.3 (only page 10)
Common errors on Typo3 web pages	No errors	1.4.4, 2.4.6
Errors on some web pages on Typo3	No errors	No errors

4.3.3 Analysis of the causes of accessibility errors

In step 6, the cause of each detected accessibility error had to be determined. In this sense, a total of four different causes for accessibility errors were detected in the analysis: the template, the HTML editor, the user and the CMS itself. Table 5 displays errors attributable to different CMS related issues. Errors made by users were not included in this table.

Table 5. Causes of the WCAG errors detected (not user related)

Checkpoint	OpenCMS	Typo3
1.3.1. Info and Relationships (Level A)	Template, CMS	Template, CMS
1.3.3. Sensory Characteristics (Level A)	CMS	-
1.4.4. Resize text (Level AA)	HTML editor	HTML editor
2.1.1. Keyboard (Level A)	Template	CMS
2.4.1. Bypass Blocks. (Level A)	Template	-
2.4.4. Link Purpose (In Context) (Level A)	Template	Template
2.4.5. Multiple Ways (Level AA)	Template	Template
2.4.6. Headings and Labels (Level AA)	-	Template
3.1.1. Language of Page (Level A)	Template	-
3.3.1. Error Identification. (Level A)	Template, CMS	-
3.3.2. Labels or Instructions (Level A)	Template, CMS	Template, CMS
3.3.3. Error Suggestion. (Level AA)	CMS	-
3.3.4. Error Prevention (Level AA)	CMS	Template, CMS
4.1.1. Parsing (Level A)	Template, HTML editor	Template, HTML editor

4.3.4 Solutions for detected accessibility errors

In step 7, solutions for found errors were proposed and they were later implemented in step 8. Due to space constraints, not all the error and the solutions found can be commented on this article. In this sense, one example is provided for one representative error of each detected cause. Errors that happened in both CMSs are selected, as they are expected to be more common in other CMSs, and, therefore, more representative of possible CMS malfunctions regarding accessibility.

- **Template (WCAG 2.4.4: Link Purpose (In Context)):**

The error was that the purpose of each link was not determined from the link text together with its programmatically determined link context. For instance, in OpenCMS it was solved by modifying the template[8] in order to allow including the title attribute for HTML tags such as <a>.

- **HTML Editor (WCAG 1.4.4: Resize Text):**

This error happens as text can be resized without assistive technology up to 200 percent without loss of content or functionality. For example, in Typo3, there are some occurrences regarding the use of italics (<i> tag) to emphasize text in web pages. In this sense, we advocate using the tag instead. In order to do so, it must be specified in its RTE HTML editor. For replacing this code, inside RTE we should select WEB, then Generated Content folder, then Footer Content folder and edit the left content . The RTE box should be deactivated and the abovementioned <i> tags should be replaced by .

- **CMS (WCAG 1.3.1: Info and Relationships):**

The information, structure, and relationships conveyed through the presentation were not available in the text. There are several types of possible situations associated with this error, such as OpenCMS form editing module not explicitly including labels related with some form elements, such as a radio and checkbox. As a concrete example, in Typo3 there are occurrences of this error caused by the Spamshield extension, which adds one invisible input in the form, filling it with a 1 if it identifies spam elements. To eliminate this input, we must go to ADMIN TOOLS → Extension Manager, click on Spamshield and deactivate the Use Honeypod box.

- **User (WCAG 2.4.2 Page Titled):**

The titles of the pages, as they are introduced by users, depend on users to describe the document correctly and fulfil the accessibility checkpoint correctly.

After step 8, following the methodology, a jump back to step 5 was performed in order to check whether all web pages were accessible in both CMS. It was found that all accessibility errors had been corrected and that the web pages were accessible. Therefore, as established in the methodology, next step was step 9, which meant that the objective of providing a proper CMS environment for developing accessible web pages that followed the same pattern of those in the sample was fulfilled.

4.3.5 ATAG re-evaluation

Finally, although not explicitly required by the methodology, an evaluation was performed to check if ATAG compliance of both CMS had improved. This evaluation also proved valid for checking whether all criteria defined in the ATAG were fulfilled.

Table 6 displays quantitative results of an ATAG evaluation of the final configuration of the CMS following the methodology.

[8] Template in OpenCMS is located in root: system/org.opencms.frontend.templateone/templates/main.jsp,

As expected, configuration changes included having raised the ATAG compliance as in Table 2. Compared to previous analysis (Table 3), 2.1, 2.2, 7.3 checkpoints are now fulfilled correctly in both CMSs.

Table 6. Quantitative results of ATAG evaluation of the final configuration of the CMS.

	OpenCMS		Typo3	
	A	AA	A	AA
Percentage of correct checkpoints	37.50 %	57.14 %	62.50 %	57.14 %

5. DISCUSSION

Proposed methodology is based on a series of steps to be performed in a concrete order with the purpose of providing a proper CMS environment for managing accessible web pages. Focusing on analyzing the accessibility of a concrete sample of web pages, it is an effective way to optimize the effort that configuring a CMS implies. In this sense, if the methodology is followed, accessibility is met for current accessibility requirements of the kinds of web pages intended to be developed by the users who employ the CMS to manage Web content. As the sample is always developed based on the specific requirements of a client or organization, only a change in the requirements would suppose modifying the CMS environments. In this sense, it is not intended to provide a fully accessible CMS environment for every kind of web page, as not all kinds of web pages are intended to be managed by the CMS environment.

It must also be taken into account that it is not necessary to fulfil the ATAG evaluation for the CMS to be able to manage accessible web pages. In this sense, by following the proposed methodology it is not intended to guarantee whether the CMS environment is accessible for users with disabilities. It must be taken into account that proper tools and plugins necessary to pass all ATAG checkpoints may not have been developed yet for a concrete CMS. Despite this fact, it is also true that the methodology can be used in order to achieve the goal of providing an ATAG-compliant CMS environment. In this sense, it provides necessary steps for evaluating the ATAG in an iterative way.

Although both CMSs are increasingly providing and incorporating improvements to facilitate the creation of accessible websites, this still does not guarantee the fulfilment of accessibility standards. Both studied CMS still have aspects to improve regarding accessibility. In this sense, the data obtained in Table 5 allows detailing the specific weak points of each CMS regarding accessibility compliance. Furthermore, from the analysis performed specific improvements have been obtained for every aspect:

- **Errors in HTML template**

HTML validation errors have been inspected and corrected in the templates, which were basically unclosed tags (in both CMSs) and appearances of "&" character in OpenCMS, which were solved by substituting them with "&amp". Language related errors in OpenCMS were corrected by modifying the template to add the "lang" attribute to the <html> tag with the specifications provided by ISO 639 [21]. Text was included in the links that appeared without text in the template. The sequence of headers was modified in Typo3 in order to avoid detected problems. Finally, as data input elements such as the search box exist without an associated label in both CMSs, the label for each element has also been introduced in the template.

- **HTML editor has not considered some accessibility aspects**

FCKEditor, although introducing <th> elements for table headers, does not associate these labels with the table cells and content, so tables are not accessible and must be manually edited to provide such association. Both HTML editors can introduce non -valid HTML content. In OpenCMS this problem is solved by using the Alkacon OAMP HTMLCleaner Module[9], while in Typo RTE the editor must be configured in order to substitute some tags by using the lib. parseFunc_RTE function.

- **CMS does not provide proper mechanisms to generate accessible code**

OpenCMS does not associate form labels and controls with "for" and "id" attributes, which causes the forms not to be accessible. The solution to this problem was to use the Alkacon OAMP Webform Module 1.3.1[10] in combination with Accessible Forms Module v0.7. Regarding webmap components, a script to generate it automatically was developed, as the CMS did not provide a way to do it by default.

- **Users can manage Web content in an incorrect way or without taking accessibility aspects into account**

There are several aspects that users can manage in a non-correct way regarding accessibility. For instance, page title management requires users to introduce correct titles for the pages regarding the content they have. Related with images, an appropriate alternative text must be provided for them. As for tables, the users must adequately define the headers that identify cells.

It is interesting that the causes for the lack of accessibility are pretty similar between both studied CMS environments. This is mainly related with the structure of the CMS, as CMS systems work in a similar way, so the problems that have arisen have similar causes. In this sense, both causes and solutions provided in the study are supposed to be helpful for other CMSs as well.

There are some helpful recommendations for users to manage accessible Web content that have arisen as possibilities that have been studied as complementary material during this work. One chance for users or systems is to provide access to an accessibility evaluation service from every web page the users manage. For instance, Moodle[11] provides this option. In this sense, users or administrators must be aware of accessibility and be able to correctly interpret the results of the report that these tools provide. It would also be helpful to use accessibility evaluation services that work inside the HTML editors employed in the CMS systems, so users would be able to check accessibility errors within the HTML editor itself. TAW for CMS[12] can be considered as a good example. Finally, although provided mechanisms would be helpful, it is necessary for users to be aware of accessibility and receive proper training regarding how to develop accessible web pages in a CMS environment.

Another aspect worth mentioning is that the accessibility analysis is focused on the characteristics in a set of pages. In this sense, all HTML elements not included in the sample have not been analyzed. These elements include multimedia related elements,

[9] Alkacon OAMP HTMLCleaner Module http://www.alkacon.com
[10] Alkacon OAMP Webform Module. http://www.alkacon.com
[11] Moodle: http://moodle.org
[12] TAW for CMS: http://www.tawdis.net/servicios/cms/

which have not been considered for this study. However, the methodology provides the necessary mechanisms for analyzing these kinds of elements. In this sense, they would be included as new requirements for the web pages to be developed and it should be analyzed whether a CMS environment provides the proper tools for developing such content. If this were the case, then it would simply imply including web pages in the sample and performing the same procedure developed in this study.

6. CONCLUSIONS

It is quite obvious that to make a CMS more accessible it should be configured properly and that the choice of a CMS alone is not sufficient for ensuring accessibility of the content for doing so in a systematic way. This paper is focused on the insights of which concrete aspects must be considered. Although some online articles provide valuable information about this [17] [18], they lack a proper methodological support to ensure the validity of their results. In this sense, the procedure provided in this paper is consistent and has strong methodological foundations as it is based on the use of sets of guidelines and the Web accessibility evaluation methodology provided by the W3C to identify the causes of the lack of accessibility in the context of CMS. Furthermore, it provides the necessary steps to find out which mechanisms are useful to ensure accessibility for a given set of representative web pages to be managed by a given CMSs. In this sense, the methodology is applicable to every CMS. It must be noted that the performed evaluation was based on a set of web pages in which not all present Web technologies were included. In this sense, it should be interesting to analyse how CMSs manage the accessibility technologies such as Rich Internet Applications and HTML5 capabilities.

This work is expected to be useful for CMS administrators to provide a proper environment for CMS users to manage accessible Web content. In this sense, it is noteworthy the growing importance that CMS are acquiring. It can also help people that manage websites to identify causes of lack of accessibility and configure CMSs appropriately for accessibility. Furthermore, it also has interest from a research point of view, as accessibility is a requirement that should be integrated into Web development methodologies. In this sense, this work can be used as part of Web development methodologies that take accessibility aspects into account.

7. REFERENCES

[1] U.S. Government, Section 508. Retrieved in March 2012 from: http://www.section508.gov

[2] Article 26 of the EU Charter of Fundamental Rights. Last accessed in March 2012 from: http://www.europarl.europa.eu/charter/pdf/text_en.pdf

[3] Chisolm, W., Vanderheiden, G., Jacobs, I. Web Content Accessibility Guidelines 1.0, World Wide Web Consortium, 1999. Last accessed in March 2012 from: http://www.w3.org/TR/WCAG10/

[4] Caldwell, B., Cooper, M., Guarino, L., Vanderheiden, G. Web Content Accessibility Guidelines (WCAG) 2.0. Last accessed in March 2012 from: http://www.w3.org/TR/WCAG20/

[5] S. Abou-Zahra. Complete List of Web Accessibility Evaluation Tools. Last accessed in March 2012 from: http://www.w3.org/WAI/ER/tools/complete

[6] Treviranus J., McCathieNevile C., Jacobs I., Richards J. Authoring Tool Accessibility Guidelines 1.0. Last accessed in March 2012 from: http://www.w3.org/TR/WAI-AUTOOLS/

[7] Richards, J., Spellman, J., Treviranus, J. Authoring Tool Accessibility Guidelines 2.0. Last accessed in March 2012 from: http://www.w3.org/TR/ATAG20/

[8] López, J.M., Pascual, A., Masip, L., Granollers, T., Cardet, X. Influence of Web Content management Systems in Web Content Accessibility. Interact 2011. Lecture Notes in Computer Science, 2011, Volume 6949/2011, 548-551

[9] Schulz, M., Pieper, M. Web Compliance Management: Barrier-Free Websites Just by Simply Pressing the Button? Accessibility and the Use of Content-Management-Systems. C. Stephanidis and M. Pieper (Eds.): ERCIM UI4ALL 2006, LNCS 4397, pp. 419 – 426, 2007

[10] Nedbal, D., Petz, G. Implementation Concept for an Accessible Web CMS. ICCHP 2010. LNCS, Volume 6179/2010, pp. 456-463

[11] Using CMS to implement accessibility and improve ROI Do current CMSs help businesses to deploy accessible sites that also provide ROI? Lofton, J. F

[12] Burzagli, L., Billi, M., Gabbanini, F., Graziani, P., Palchetti, E. The Use of Current Content Management Systems for Accessibility. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2004. LNCS, vol. 3318, pp. 331-338

[13] Xiong, J., Farenc, C., Winckler, M. Analyzing Tool Support for Inspecting Accessibility Guidelines during the Development Process of Web Sites. In Proceedings of 1st IWWUA workshop inside WISE 2007. Springer LNCS 4832. Pp. 470-480.

[14] Brajnik, G.: Web Accessibility Testing: When the Method is the Culprit. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 156–163. Springer, Heidelberg (2006)

[15] Abascal, J., Arrue, M, Vigo, M. A Methodoloy for web Accessibility Development and Maintenance. In: Zaphiris, P., Kurniawan, S.: HCI Research in web Design and Evaluation. Idea Group Publishing, 2006. ISBN. 1-59904-248-7.

[16] López, J.M., Pascual, A., Granollers, A. Engineering Accessibility in Web Content Management System Environments. WISE 2009. LNCS Vol. 5802, pp. 597-604

[17] Connor, J.O. Choosing an Accessible CMS. Retrieved in March 2012 from: http://juicystudio.com/article/choosing-an-accessible-cms.php

[18] NCDAE Tips and Tools: Content Management Systems & Accessibility. Retrieved in March 2012 from: http://ncdae.org/resources/factsheets/cms.php

[19] Evaluating Web Sites for Accessibility. Retrieved in March 2012 from: http://www.w3.org/WAI/eval/Overview.html

[20] Hezart, A., Naik, S., Araujo, A., Watters, P.A. "Solving Frame-Based Accessibility Problems in Web Content Management," CIMCA-IAWTIC'05, vol. 1, pp.246-250, 2005

[21] ISO 639: Code for the Representation of Names of Languages. 1st ed. Geneva: International Standardization organization, 1988. 17 p

[22] Royal decree number 1494/2007. Spanish. Official State Bulletin 279, 2007, pp. 47567– 47572 (In Spanish). Last accessed in March 2012 from: http://www.boe.es/aeboe/consultas/bases_datos/doc.php?id=BOE-A-2007-19968.

[23] Italian Government. Requisiti tecnici e i diversi livelli per l'accessibilità agli strumenti informatici. July 2005. G. U. n. 183 8/8/2005. Last accessed in March 2012 from: www.pubbliaccesso.it/normative/DM080705.htm.