Journal of Web Engineering, Vol. 12, No.3&4 (2013) 181-202

SUPPORTING NAVIGATION ACCESSIBILITY REQUIREMENTS IN WEB ENGINEERING METHODS

LOURDES MORENO[1], FRANCISCO VALVERDE[2], PALOMA MARTÍNEZ[1] and OSCAR PASTOR[2]

[1]*Computer Science Department*

Universidad Carlos III de Madrid, Madrid (Spain)

{lmoreno, pmf}@inf.uc3m.es

[2]*Research Center on Software Production Methods (ProS)*

Universitat Politècnica de València, Valencia (Spain)

{fvalverde, opastor}@pros.upv.es

Received September 21, 2011

Revised January 7, 2013

Web accessibility not only guarantees universal user access to the Web, but also provides interesting benefits for Web development. In order to achieve the goal of Web accessibility, an interesting approach is the incorporation of accessibility requirements into current Web engineering methods. This article presents the Accessibility for Web Applications (AWA) approach with the aim of integrating accessibility into Web engineering methods. The paper also discusses the application of the AWA approach to the Object-Oriented Web Solutions (OOWS) engineering method to produce accessible Web applications with a focus on navigational requirements. In order to demonstrate the practical applicability and usefulness of the approach, a proof of concept is described, the results of which indicating the satisfaction of navigation accessibility requirements. With the application of the AWA approach in the model-driven development (MDD) method, previously-defined OOWS models have been extended with the accessibility criteria, providing resources for the required changes in the process.

Key words: Web accessibility, Web engineering, Design

Communicated by: G.H. Houben & G. Rossi

1 Introduction

It is no longer the case that the only users affected by barriers to Web accessibility are individuals with disabilities. Indeed, as many users experience problems using the Web resulting from not only disabilities, but also functional and technological limitations, as well as other environmental factors, it can be said that the digital divide is, in fact, growing.

One important demographic trend to consider for Web accessibility is the aging general population and work force (i.e., many workers are choosing to retire at an older age) in industrialized countries. As many of these older workers use the Web not only for work-related and administrative tasks, but also for leisure activities, e-government, e-learning, socializing, banking and information access, the result may be the increased importance and number of age-related accessibility barriers encountered by older users. In order to avoid the exclusion of these individuals, as well as many others, from the Web, the goal of e-inclusion must be vigorously pursued. Considering the possible negative economic ramifications of accessibility barriers for this growing percentage of the workforce, one sees Web accessibility as a goal to be pursued on more than just social justice grounds alone.

In order to design a universally accessible Web, important initiatives exist for the legislation, standardization and regulation of accessibility guidelines and standards. Nevertheless, current statistics show that these directives are often not followed and barriers to accessibility continue to exist today [27].

One of the main reasons for this is the lack of proper knowledge and training in the IT community on matters related to Web accessibility. Misconceptions on the topic have resulted in disproportionate attention being paid to the difficulties of implementing accessibility standards in Web development; while the advantages and economic benefits derivable from accessible Web development have received scant attention [11].

The authors of this paper take accessibility as a criterion for quality to be guaranteed in all Web development projects. While it is true that the incorporation of accessibility criteria in early Web development phases may be linked with higher cost predictions and longer development processes, these costs become significantly higher, to the point of rendering the endeavour virtually impracticable, when accessibility is taken into account at later stages. It has been the perception of Web developers that methodologies are needed which incorporate Web accessibility throughout the entire development process [19].

It is interesting to note that, despite endeavours such as the Web Content Accessibility Guidelines (WCAG) [45], very few approaches offer support for the incorporation of accessibility requirements in the software development process. Instead, where accessibility requirements have been dealt with, little discussion can be found in the literature – despite the existence of a few articles on particular development activities – on how their incorporation should be carried out. Rather, the works indicate a growth in Web application evaluation technique research and highlight the lack of research attention dedicated to numerous development activities [12].

In an attempt to rectify this lack of attention to accessibility issues in Web development, Accessibility for Web Applications (AWA) was created as a methodological framework supporting the development of accessible Web applications from the perspective of Web engineering [23]. With accessibility defined as the fulfilment of WCAG, the AWA approach takes as its overall goal the definition of specific accessibility requirements to be included at different steps of a Web engineering method such that overall accessibility may be ensured. These requirements have been defined with a degree of independence sufficient for their use in different Web engineering methods.

AWA includes conceptual elements capable of abstracting WCAG accessibility requirements, as well as patterns that allow for mapping between conceptual primitives and accessible HTML code. Of course, it is important to mention that, as not all accessibility requirements can be dealt with in conceptual modelling and some must be addressed in implementation, implementation guides are also provided by the AWA approach. Nevertheless, this latter feature lies outside the scope of the present article.

To demonstrate the power of AWA in the Object-Oriented Web Solutions (OOWS) engineering method [10] and according to an MDD strategy, a subset of accessibility requirements related to navigation has been selected here. Section 2 of the paper discusses related work, while the AWA methodological framework is presented in Section 3. In Section 4, the AWA approach for Web engineering methods and the Web navigation metamodel are presented. Section 5 offers a proof of concept for the AWA approach in the OOWS method, as well as the process whereby accessibility requirements are incorporated therein. Finally, Section 6 offers concluding remarks and proposes areas for future research.

2 Current Practice and Research

In order to achieve comprehensive Web accessibility, a significant number of initiatives, legislation and standards exist which identify problems and suggest new, accessible designs. Among accessibility standards, the World Wide Web Consortium (W3C) along with the Web Accessibility Initiative (WAI) both deserve special mention [42]. The Web Content Accessibility Guidelines (WCAG) [45] are the most important component of the WAI and for which two versions currently exist, namely, WCAG 1.0 and WCAG 2.0. While the former version is still that named in many legislative and regulatory frameworks, in other contexts it has been supplanted by WCAG 2.0 since as early as its publication date as a W3C Recommendation in December of 2008. In the European Union and following Digital Agenda and the standardization mandate 376 [6][9], WCAG 2.0 is considered the official standard. WCAG 2.0 is also referenced in the legislation of many other countries. Australia, Canada, Hong Kong, Japan and New Zealand, for example, have already adopted WCAG 2.0. Other important WCAG 2.0-based initiatives include the recent ISO/IEC DIS 40500 [17], BITV 2 [5] in Germany, RGAA [32] in France, AODA [28] in Ontario, JIS X 8341-3 [16] in Japan, UNE 139803 in Spain [1] and Section 508 (29 U.S. Code § 794d) in the United States [33]. Although less extensive, these standards are nevertheless very similar to the WCAG and studies can be found comparing the former and latter.

WCAG 2.0 follows an approach different from that of WCAG 1.0, inasmuch as the former was developed to be applied not only to W3C technologies, but also to other current and emerging technologies. As will be seen in the following sections, the present paper and the techniques discussed herein are based entirely on WCAG 2.0 [43].

Of the number of relevant studies found based on WCAG 1.0 and 2.0, the majority deal with the evaluation of accessibility, be it the quantitative measurement of accessibility [36] or qualitative evaluation with necessary tools [1] [44]. While some authors offer instructions for the use of these guidelines [34] or propose simple frameworks in an attempt to maximize benefits from guideline implementation [26], very few address the question of how to incorporate accessibility requirements in the software development process [20]. In the studies which do, piecemeal suggestions are generally

offered rather than comprehensive solutions [4]. There has also been a notable lack of technological support in the different activities taking accessibility into account [38].

Some studies have been found proposing the application of WCAG to semantic Web technologies like ontologies, XML, XMLSchema and OWL [21][30]. Among studies found in the area of Web engineering methods, some propose the use of patterns in a method for the inclusion of particular accessibility requirements in user interfaces [18], while others propose the use of the WebML method, focused on user-friendly and efficient access in navigation and information retrieval [3].

Nevertheless, very few articles have been found which directly address the question of how to model accessibility according to WCAG and only one was found with an aspect-oriented approach [20]. In this way, none of the previous approaches fully satisfy accessibility standards. An interesting attempt meriting particular mention, however, is the DANTE approach integrating the Web Authoring for Accessibility (WAfA) ontology [37][14] for the visually impaired into WSDM method [30]. This latter approach has similarities with the AWA approach presented here. Both, for instance, make an abstraction of accessibility concepts for the integration of accessibility through the inclusion of navigational and structural semantics. To automatize this process, both approaches propose an MDD strategy, integrating the requirements in a Web engineering method. Furthermore, both define the relationship between accessibility concepts and method requirements or modeling concepts as a strategy. Nevertheless, important differences between the two approaches do exist. Whereas DANTE's focus with the use of the WAfA ontology does not provide WCAG 2.0-based concepts, the AWA approach includes WCAG 2.0 requirements. While it is true that there is some overlap in some concepts considered by both approaches, this is not the case throughout. Of particular interest in this regard is that WCAG 2.0 is a product of consensus and, as such, offers benefits to distinct groups of users with disabilities. AWA, therefore, incorporates aspects of accessibility not considered in the DANTE approach with the WAfA ontology. While DANTE approach accessibility requirements, for example, permit semantic annotation to provide screen readers with extra information to better facilitate the audio presentation of a web page, DANTE does not consider other scenarios or modes of access beyond that of screen readers.

3 AWA Approach

Accessibility for Web Applications (AWA) provides a framework with methodological support for the inclusion of accessibility requirements in the Web application development process [23]. AWA provides guidance to Web engineers for the incorporation of the following types of accessibility requirements from different perspectives: (a) organization and business-related requirements that integrate accessibility and quality policies (e.g., an accessibility training plan in the company or the selection of technology used in accessibility evaluation), (b) requirements regarding the Web development method and following a methodological approach for their systematic integration – following WCAG – from the beginning of the process, and (c) a user-centred design (UCD) approach anticipating user participation in the design process [15][22].

In each case, the full scope of functional diversity and unfavourable contexts of use on the Web is considered from the vantage point of universal design [25]. Tailoring the AWA approach to these

different perspectives, three different AWA support components – Organization, WCAG and Interaction – were developed.

These components were then applied to a real-world scenario, as explained in later sections of this article. Section 4 discusses the integration of the WCAG component (focusing on navigation-oriented requirements) in Web engineering methods following a MDD strategy. This is concretely demonstrated in Section 5 in the OOWS engineering method. The result of the application of AWA, as will be discussed in much greater depth, is the accessibility (i.e., accordance with WCAG) of the resulting Web pages.

For the development of the three components, as well as the set of requirements, various sources including WCAG, ISO 13407, the more recent ISO 9241-210 and human-computer interaction (HCI) methods were consulted in order to follow UCD approach. With this diverse set of standards, a study was performed to determine how and when exactly they could be incorporated in the early stages of the development process. In order to ensure the quality of the accessibility provided, a classification of requirements was obtained.

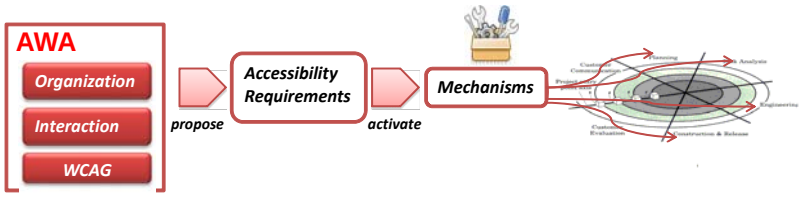

Figure 1 The AWA methodological approach.

The requirements activate a number of mechanisms which, in turn, offer resources to professionals for the application of the methodological support framework. It is these specific mechanisms which are responsible for running the accessibility requirements in each phase of the development process. Figure 1 represents the three AWA components including related requirements. The ‘propose’ arrow represents the conceptualization of requirements from different sources, while the ‘activate’ arrow represents the different mechanisms responsible for fulfilling requirements. The mechanisms are associated with a given activity at a specific moment in each Web development process.

The AWA approach aims to offer developmental support for the generation of accessible web pages and was created for application in Web engineering methods for the automatic production of Web applications, as well as Web applications generating dynamic websites (i.e., database-backed sites with dynamically generated content). These destination approaches must make use of client-side technologies like W3C standards (e.g., XHTML, HTML and CSS) and server-side technologies. That said, the support of rich internet applications and the inclusion of accessibility requirements according Accessible Rich Internet Applications (WAI-ARIA) currently fall outside the scope of AWA [24].

3.1 Accessibility requirements of WCAG component

As shown in Figure 1, the WCAG component is abstracted from official WCAG documentation semantics. WCAG, as mentioned earlier, is a widely-accepted standard for the definition of Web accessibility focused on application during evaluation, rather than in the development process. The

authors analysed official WCAG documentation semantics, as well as those of other WAI documents, obtaining through abstraction implicit accessibility requirements of different natures or types. This was done in part to guarantee that accessibility requirements be incorporated at different points of the software development process. The goal is referred to here as a quality requirement and is pursued through the systematization of accessibility for each requirement, such that it can be addressed in the design. The requirements classification obtained distinguishes between navigation, content, presentation and user interaction requirements [23]. Some requirements can be included in the design phase, whereas requirements that cannot be tackled in the modelling phase must instead be addressed in the implementation phase. As mentioned earlier, AWA also provides implementation guides to support these requirements.

While the abstraction is performed in the context of the software development process, these requirements nevertheless maintain a direct and complete correspondence with WCAG (Table 1 of sub-section 4.1 illustrates the correspondence between WCAG 2.0, WCAG 1.0 and the navigation-related requirements).

3.2. How is the WCAG component applied?

As described, the AWA support framework application incorporates accessibility requirements through the activation of different mechanisms associated with a given activity at a specific moment of the Web development process.

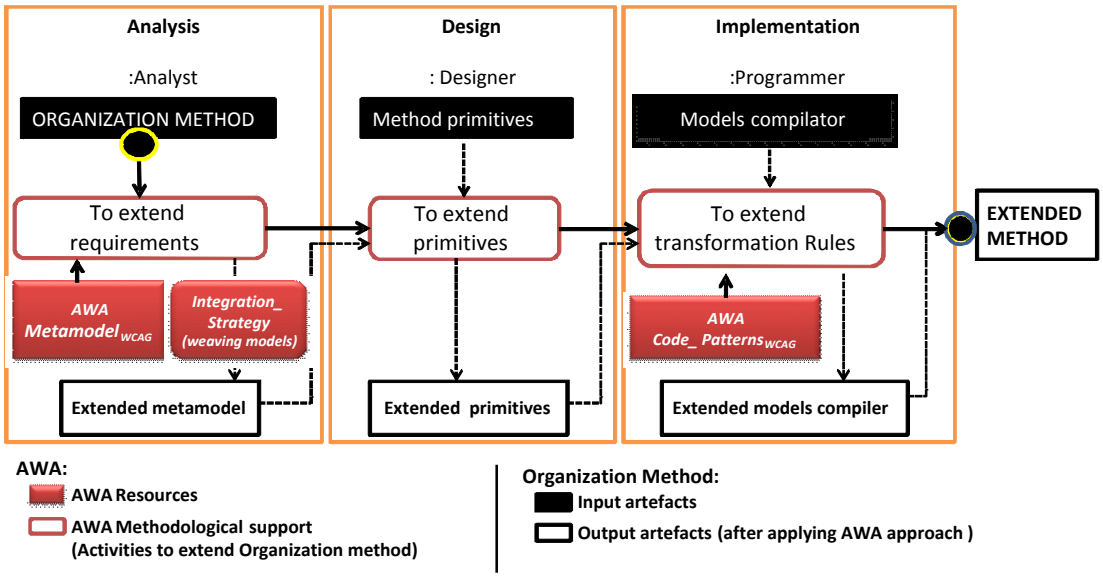

Figure 2 Activity diagram of Step 1, 'Extend organization method', applying AWA approach.

If an organization wants to apply the AWA approach to develop accessible Web applications, it must follow two principal steps:

- **Step 1: Extend organization method.** A software production organization follows a specific development method. This method could follow a more or less agile approach, as well as an MDD strategy. The figure 2 shows this step with an UML activity diagram. AWA provides a metamodel that incorporates the accessibility requirements concerning WCAG (denoted by $AWA\_Metamodel_{WCAG}$ in Figure 2). This metamodel must be integrated in the method metamodel (see Figure 2, 'Extended Metamodel' and 'Analysis' phase). To conduct this integration, different development method-dependant integration strategies are followed. The metamodel defined in AWA is described using MOF and OCL languages.

  Figure 2 shows the activity diagram of this approach. In the design phase, primitives are extended to deal with the extended metamodel. Finally, in the implementation phase, the transformation rules are also extended with the use of AWA-provided code patterns (denoted by $AWA\_Code\_Patterns_{WCAG}$ in Figure 2). The final result is an organization method extended to include accessibility requirements in the metamodel. This step is taken once in the organization method. The approach applied to the OOWS engineering method is shown below in Section 5.

- **Step 2: Web application development.** The accessibility requirements that have not been included through modelling using the extended method must be integrated by other means for each new Web application. These requirements are related to the concrete presentation and accessibility evaluation process.

4 Approach for Web Engineering Methods

This section demonstrates how accessibility requirements are integrated in conceptual modelling following the 'extend organization method' approach described previously. As shown in Figure 2, tree resources have to be defined including the metamodel, the patterns to translate conceptual primitives into HTML code, as well as an integration strategy to map the metamodel in the Web engineering methods:

- **AWA_Metamodel.** For each guideline in WCAG, a conceptual model is defined to represent a given requirement. The main goal of this metamodel is the description of each accessibility concept that a model must include. The metamodel was defined generically enough so as to be method-independent.

- **Integration_Strategy.** In order to integrate the AWA_Metamodel into current Web engineering methods, an integration strategy is proposed. First, the method modelling entities which must support an accessibility requirement are selected. Next, a strategy is proposed according to which relationships are created between these modelling entities and the accessibility metamodel entities describing the additional information. In order to capture such relationships, weaving models have been applied [7]. A weaving model establishes the relationships between the modelling entities of two models that are not directly related. Through the application of this strategy, changes in the method metamodel are avoided.

- **AWA_Code_Patterns.** These patterns describe the accessible HTML code to be generated for each conceptual primitive from AWA_Metamodel. The patterns are used to improve the code generation phase of the MDD strategy.

Accessibility requirements that cannot be tackled in the modelling phase must be addressed in the implementation phase. While the AWA approach also provides implementation guides to support these requirements, this lies beyond the scope of the present article.

4.1 AWA_Metamodel

As the focus of the present article is navigation-centred accessibility requirements, the following WCAG 2.0 guidelines have been selected:

- Guideline 1.3: "Create content that can be presented in different ways without losing information or structure."
- Guideline 2.4: "Provide ways to help users navigate, find content and determine where they are" with success criteria including "the purpose of each link", "multiple ways", "Web pages have titles that describe topic or purpose" and "headings and labels describe topic or purpose".

From the analysis of these guidelines, the following accessibility requirements must be addressed. These requirements are illustrated in Figure 3.

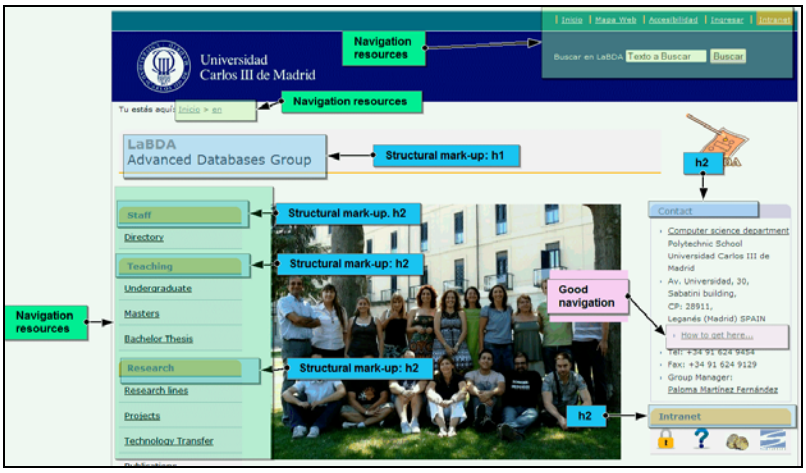

Figure 3 Screenshot of web page with points indicating accessibility requirements for navigation.

- Structural mark-up: Web contents must be logically organized using HTML heading elements (h1, h2, h3, h4, h5, and h6). Additionally, web pages must have titles that describe their topic or purpose.
- Good navigation: The purpose of each link and its labels must be clear to the user.
- Navigation resources: In order to locate the Web content, navigation mechanisms (e.g., breadcrumb trail, site map and navigation bars) as well as current location in the application must be provided to the users.

Table 1 shows WCAG 2.0 Success Criteria supported with each of these requirements. It is noteworthy that only 8 of the 61 WCAG 2.0 Success Criteria have been dealt with here. However, Level A Conformance can be reached using these selected success criteria. This justifies the fact that the requirements must be systematized from early stages of the development process.

Table 1. Mapping between guidelines (WCAG 1.0 & WCAG 2.0) and accessibility requirements[1] with HTML Techniques for WCAG 2.0.

WCAG 1.0 Checkpoint / Priority		WCAG 2.0 SUCCESS CRITERIA / Level Sufficient Techniques WCAG 2.0			$AWA\_Requirement_{WCAG}$ Name	$AWA\_Requirement_{WCAG}$ Code
3.5	2	1.3.1	Info and Relationships Techniques: H42, ARIA1, ARIA4	A	Structural mark-up	N 01.02
12.1 (*)	1	2.4.2	Page Titled Techniques: G88, H25, ARIA1	A		
13.8 (*)	3	2.4.6	Headings and Labels Techniques: G130, G131	AA		
3.5 (*)	2	2.4.10	Section Headings Technique: G141	AAA		
13.1 (*)	2	2.4.4	Link Purpose (in context) Techniques: G91, G53, ARIA1	A	Good navigation	N 01.01
13.1 (*)	2	2.4.9	Link Purpose (link only) Techniques: G791, C7	AAA		
13.3	2	2.4.5	Multiple Ways Techniques: G125, G63, G64, G126	AA	Navigation resources	N 01.03
13.9 (*)	3	2.4.8	Location Techniques: G63, G65, G128, G127, H59	AAA		

*No direct WCAG 2.0 equivalent

Figure 4 shows the proposed metamodel for the collection of the navigation-related requirements according to WCAG. To support the requirement, *Structural Markup* (*N 01.02*), the *AccessiblePage*

1 Comparison of WCAG 1.0 Checkpoints to WCAG 2.0 (http://www.w3.org/WAI/WCAG20/from10/comparison/)

entity, represents a Web page. The topic attribute describes the contents of the page having a correspondence with the text to include in the Title HTML element in the final Web code required to achieve WCAG 2.0 Success Criteria 2.4.2.

An *AccessiblePage* is made up of several *ContentNodes*, which are also described using the topic attribute and used to describe the HTML heading elements (h1, h2, etc.) to achieve WCAG 2.0 Success Criteria 1.3.1. Additionally, if the topic attribute provides descriptive headings, the technique G130 can be applied and WCAG 2.0 Success Criteria 2.4.6 can be fulfilled. The *ContentNodes* are structured into different levels using the Header entity. WCAG 2.0 Success Criteria 2.4.10 indicates that the headers organize the content, according to the technique G141. The level association between *ContentNode* and *Header* establishes the ordering sequence between different headers. As an example, the first header is considered as the title of the content, whereas subsequent headers represent the sub-sections. Using these modelling entities, an organized hierarchy of webpage contents can be described.

The requirement *Good Navigation* (*N 01.01*) is supported by the *purpose* attribute of the *Link* modelling entity. This attribute has a correspondence with the text of the link purpose in the final Web code (as described in 13.1 of WCAG 1.0, as well as 2.4.4 and 2.4.9 of WCAG 2.0) using the technique G91, indicating that the link text must describe the purpose of a link.

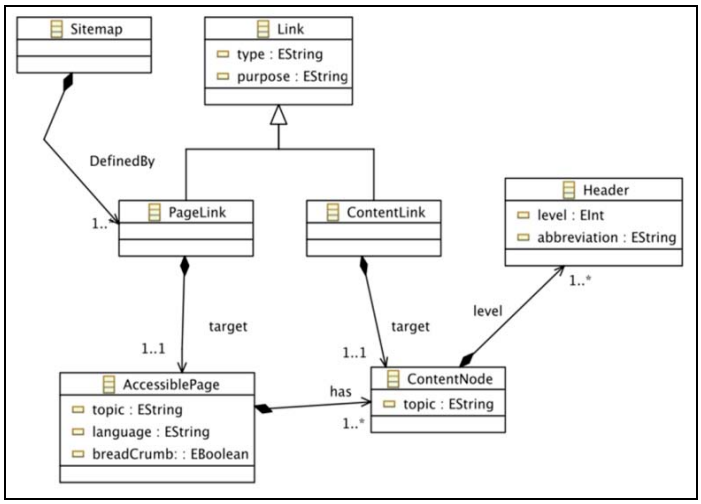

Figure 4 AWA_Metamodel corresponding to navigation requirements.

Finally, as regards the requirement *Navigation Resources* (*N 01.03*) for the location of Web content, WCAG 2.0 Success Criteria 2.4.5 indicates that more than one way is required to locate a web page and using two or more of the many techniques proposed by WCAG 2.0: (a) providing a site map according to technique G63 represented with the *Sitemap* entity made up of the different *PageLinks* and (b) the Boolean *breadcrumb* attribute of the *AccessiblePage* entity has a correspondence with the breadcrumb trail navigation indicated in the technique G65 to achieve WCAG 2.0 Success Criteria 2.4.8. If the attribute is set to *true*, the current location of the user must be shown in the web page.

4.2 AWA_Code_Patterns

Providing a helpful resource to programmers, AWA_Code_Patterns is a mechanism activated by an AWA_Metamodel meta-element, as described before. The pattern indicates the expected final code as a result of introducing the requirements described in the metamodel.

```
<a href=”Link.URL”>Link.purpose</a>
```

Figure 5 AWA_Code_Pattern N 01.01 (good navigation).

Figure 5 shows a patterns for requirement *N 01.01* indicating what the resulting code would be if a *Link* element were included. This *Link* element is described from metamodel entities (see Figure 4). This requirement indicates that the text or anchor link of the *Link* element must describe the purpose of the link.

```
<doctype…>
<html>
    <head>
        <title>AccessiblePage.topic</title>
        …
    </head>
    <body>
        Process_all_asociated_ContentNode
    </body>
</html>

Process_all_asociated_ContentNode:
Foreach (ContentNode)
 {
   level = Header.level
   if(level == 1) {
      if(existH1) {
         warning(“two h1 shouldn’t exist)
      }
      <h1>topic</h1>
      existH1 = true
      }
   else if(level == 2)
      {
      <h2>topic</h2>
      }
   …
   }
```

Figure 6 AWA_Code_Pattern N 01.02(structural mark-up).

In relation to requirement *N 01.02*, a pattern is shown in Figure 6 indicating both that a website should have *Header* elements and how the header should include attributes from the *ContentNode* entity of the metamodel described. For requirement *N 01.03*, a pattern is not offered due to the multiple coding possibilities present.

5 Proof of Concept: Applying AWA to the OOWS Engineering Method

OOWS is an engineering method providing methodological support for Web application development. The method was developed as an extension of the OO-Method [19] in order to provide better support for Web-related concepts. In order to achieve this goal, OOWS introduces a new set of models (a user model, navigational model and presentation model) to support the interaction between the user and a

Web application. However, several accessibility barriers had been detected when a Web application was generated using the OOWS method. For that reason, the method was selected as a proof of concept for the AWA approach.

In order to evaluate the approach, two tasks were performed. First, the OOWS engineering method was extended using AWA, as described in Section 4. This extension was carried out at the levels of modelling (sub-section 5.3) and code generation (sub-section 5.4). Secondly, the authors analysed the expected improvement of the accessibility in the modelling of a case study. A lab demonstration is presented here using a case study as empirical validation research method. The demonstration justifies that the methodology could be used in practice and shows the utility of the proposed approach, offering an explanation in terms of mechanisms used to include accessibility requirements in software development (sub-section 5.5) [40][41].

5.1 OOWS

In order to support Web application development, OOWS includes a navigational model to gather navigational requirements that is made up of several (sub-)models. First, users that directly interact with the system are represented in a user diagram. Next, the abstract tasks that the users can carry out with the system (e.g., adding a new customer to the system or browsing a product catalogue) are represented as 'navigational contexts'. The navigational contexts accessible to a specific user are described using a navigational map. Navigational contexts are classified as 'first level' if they are always accessible to the user or as 'sequence' if they are only accessible by means of navigation from another context. The navigational map also supports grouping a set of contexts together as a 'subsystem'.

Navigational contexts are further decomposed into abstract interaction units (AIUs) which specify an atomic interaction (e.g., the retrieval of a set of data objects). An AIU is defined as an interaction view over the state of the information system objects as defined in an object model. Two types of AIUs exist, namely, (1) a population AIU which retrieves a set of data for interaction and (2) a service AIU which represents an interaction for a service execution from a set of input arguments.

To specify the information to be shown, each population AIU is made up of a set of stereotyped classes with the «view» keyword that represent views over the attributes of the system objects. Each population AIU has one mandatory class, called a manager class, and a set of optional classes, called complementary classes, to provide complementary information from the manager class.

5.2 Case study: 23andME

In order to validate the AWA approach, justify the usefulness of the proposal and analyse the accessibility issues of a Web site described using OOWS Models without including the AWA extension, we present a laboratory demonstration. The web site modelled is 23andMe, a personalized genomics web portal. This portal offers personal genomic service from a user-provided DNA sample, analysing 966.983 SNPs of all human chromosomes. From the results of this analysis, they provide information about health, ancestry and the user's genome. 23andMe is a good example of current Web 2.0 trends, as it has been designed to provide genomic information in an understandable way for end-

users and, therefore, includes collaborative and social features. While the full description of all the details about the 23andMe modelling process lies outside the scope of this paper, further details can be found in another study [13].

To perform the evaluation, the first step was to carry out an accessibility evaluation of navigation requirements using the WAVE Tool [39].WAVE provides a technical report and shows the original web page with embedded icons and indicators that reveal the accessibility of the page (see Figure 7). To carry out this analysis, the public view of the 23andMe commercial site was used. As OOWS models represent the same navigational and content structure, navigation accessibility issues detected in the website are also present in the Web application defined using OOWS.

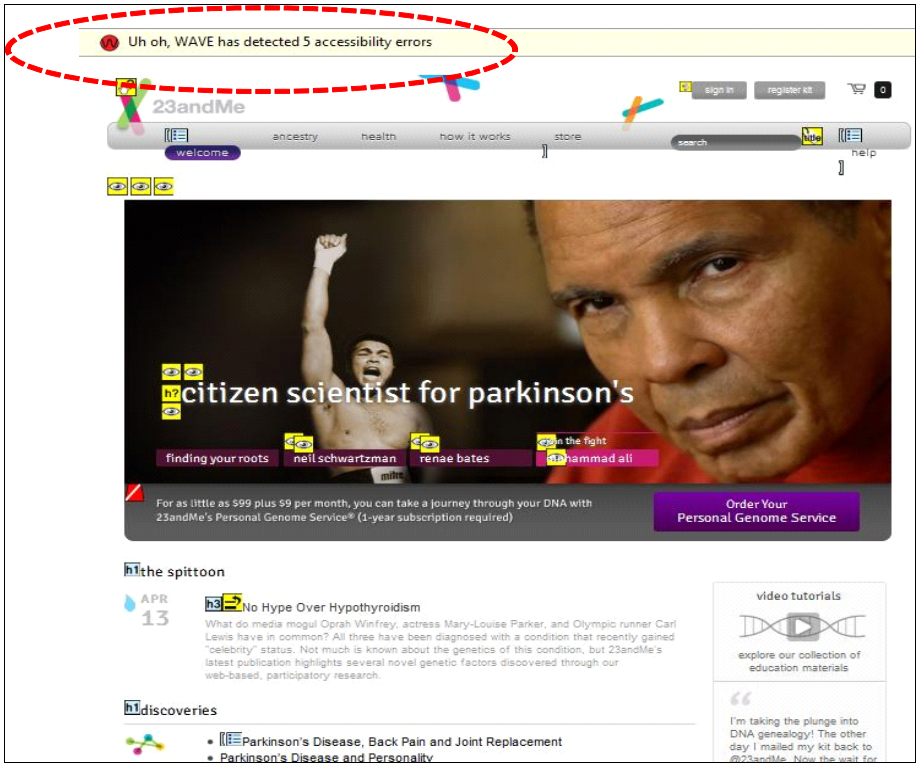

Figure 7 Display page of WAVE Tool when checking 23andMe web page.

The accessibility report generated detected that requirements N 01.02: Structural Mark-up and N 01.03: Navigation Resources are not supported. Both errors are present in nearly all the website pages analysed since navigation is a concern widely present in any Web application. Furthermore, requirement N 01.01: Good navigation is supported by the Web application, as highlighted in Figure 8, but not by the OOWS modelled application. Due to limitations on the modelling level, it must therefore also be considered in the extension.

Regarding the structural mark-up requirement, while heading elements are present, they are incorrectly ordered, as shown in Figure 9. To establish a logical ordering, OOWS models must support

a hierarchy of the information shown. This hierarchy is implicit in how information views are modelled: manager classes always show information related with the "h1" HTML element, whereas complementary classes show the information related with the subsequent headers (see UML comments in Figure 10). In this latter case, UML comments have been added to the OOWS modelling notation (see Figure 10) with the purpose of defining a logical ordering.

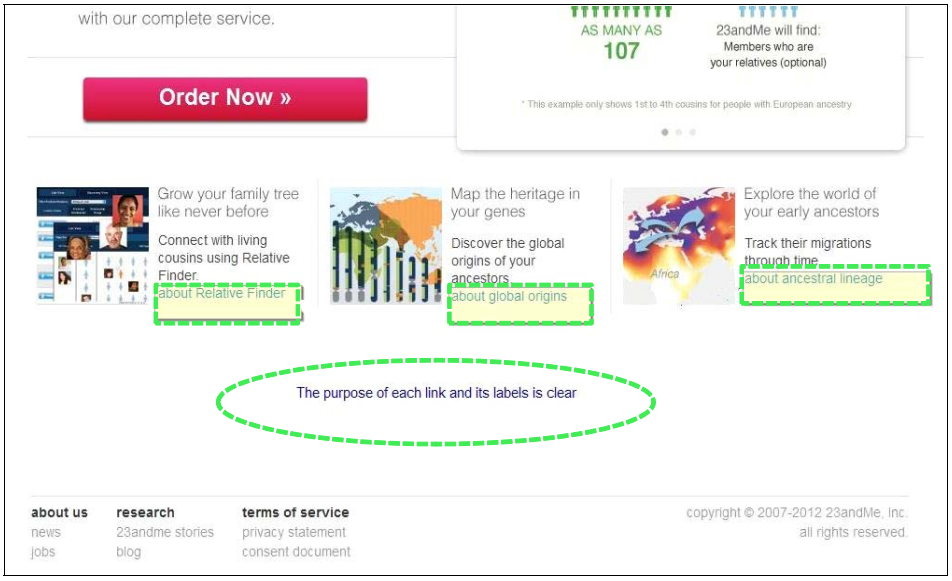

Figure 8 23andMe web page demonstrates good navigation (requirement N 01.01).

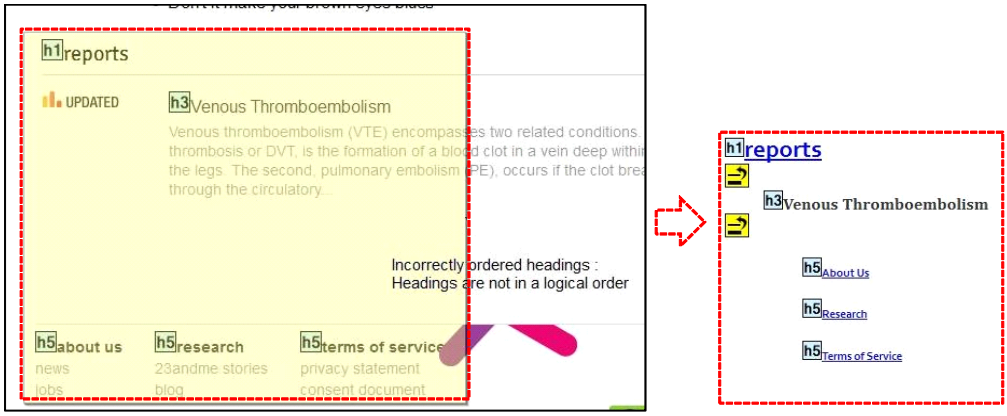

Figure 9 23andMe web page incorrectly shows ordered heading elements.

Regarding the navigation resources requirement, the issue detected was that OOWS models provide only a single resource for navigation: a main menu or navigation bar generated from the

information provided by the navigational map. To solve this issue, two additional mechanisms, breadcrumbs and site maps, must be generated from OOWS models.

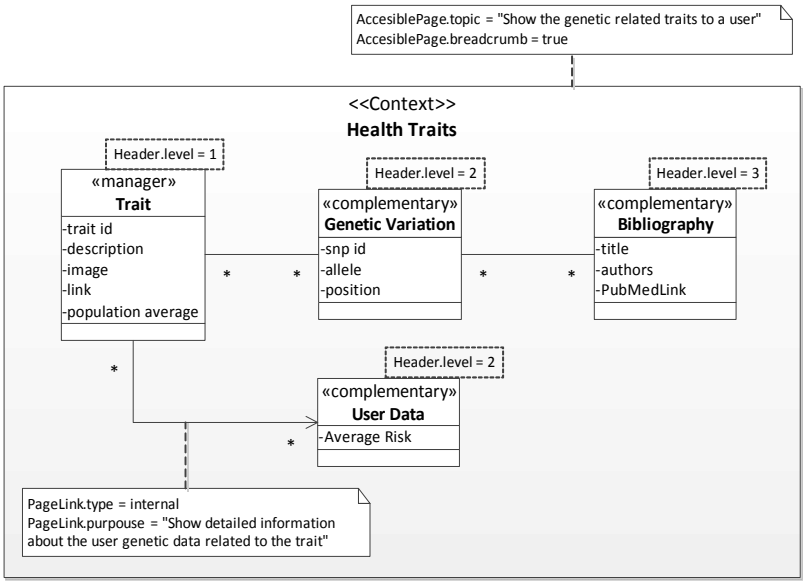

Figure 10 Extended OOWS model with accessibility information in 23andMe.

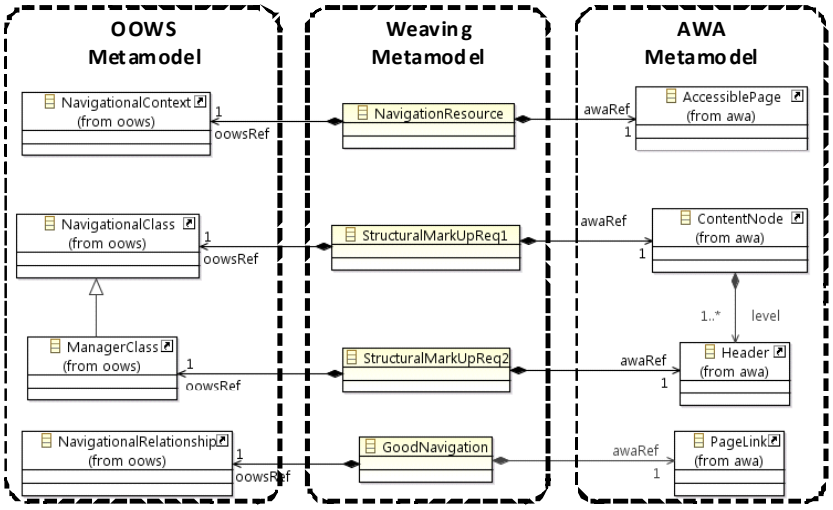

Figure 11 Weaving metamodel between OOWS and AWA.

5.3 Metamodel extensions

In order to respond to the accessibility requirements of the OOWS method, both the AWA_Metamodel and the AWA_Integration Strategy have been applied. To perform the integration, a weaving

metamodel between the OOWS metamodel and the AWA_Metamodel was defined [7]. In this new metamodel, each OOWS modelling entity translated into non-accessible code was related with the corresponding AWA_Metamodel entity. Thus, when a weaving model is defined, the OOWS conceptual model is extended with the expressivity of the accessibility required to conform to the WCAG standard.

In order to support the navigation-related accessible requirements presented in Section 4, the relationships defined in the OOWS-AWA weaving metamodel are detailed below. Figure 11 shows the set of relationships defined between the entities of the OOWS and the AWA_Metamodel. To illustrate these relationships with an example, Figure 10 presents an OOWS model for showing the health traits described in 23andMe that is extended with accessibility requirements (represented as a note attached to the corresponding modelling entity):

- **Requirement *N 01.01 (good navigation).*** A manager class can be related to a complementary class by means of a navigational relationship. This relationship implies a navigational capability to a target navigational context, creating a sequence link in the navigational map. Figure 10 illustrates a navigational relationship between 'trait' and 'user data' depicted with a solid arrow. The generated links from this modelling entity only provide the name of the target web page. To support accessible navigational links, each navigational relationship is related to the *PageLink* AWA entity in order to define the link purpose. The attribute type is also set to internal by default since OOWS does not define external links.
- **Requirement *N 01.02 (structural mark-up).*** Different navigational contexts are described as an aggregation of one or more AIUs which represent the requirement of retrieving a chunk of related information. That information is represented by the attributes of several navigational classes. Each AIU must include one navigational class, called the manager class, and optionally include a set of complementary classes to provide additional information.

  Each navigational class must be related to a *ContentNode* of the AWA metamodel in order to use the topic attribute for describing the content. Currently, the information retrieved is shown to the user without providing a logical structure or in a single level of content. To respond to this situation, manager classes are always linked to a *Header* entity that represents the first content level. Additionally, a *Header* entity can be linked to one or more complementary classes to show their information in a specific sub-level. According to Figure 10, information about the genetic variations associated to a trait and the user data will be shown in the second level, whereas bibliographic information will be shown in the third.
- **Requirement *N 01.03 (navigation resources):*** In OOWS, the navigational map defines the system's navigational structure for each type of user. This map is depicted by means of a directed graph whose nodes represent navigational contexts and whose edges represent navigational links defining valid navigational paths. In OOWS, each webpage is generated from a navigational context. For that reason, the AWA *AccessiblePage* entity is related to a context in Figure 11. By default, the OOWS code generation process creates a breadcrumb for every web page. However, there is no a mechanism to define a site map. To solve this issue,

several *Sitemap* AWA entities are related to the different navigational maps. Hence, all links that define the application access can be obtained to define the *Sitemap* for each user.

Hence, the OOWS conceptual model is extended with the required accessibility information. The next step is to modify the corresponding code generator in order to support the specific AWA_Code_Patterns. This improvement must be done manually, as each method has its own specific implementation. However, the proposed patterns provide an effective guide and guarantee that corresponding guidelines from WCAG are complied.

5.4 Code Generation Extensions

With the aim of supporting accessible code generation, the OOWS M2C transformation process, based on XPand language rules [8], was redefined. This transformation process translates the information from the OOWS models to a custom PHP framework that generates the expected HTML. Details about this framework can be found in another study [35]. To establish the new code generation process rules, the corresponding AWA code patterns were considered. First, Figure 12 shows a code excerpt with the modifications to support the structural mark-up requirement. Mainly, when a new chunk of HTML information is generated using the AddInformationZone method (line 3), the header level is also the first method parameter. Next, the rule NavRelationship_rule, which generates the HTML related with the complementary information, is called recursively (line 4). In this rule, the information from complementary classes is generated using the ZoneDetailRelationship method (line 11). Taking into account the header level parameter, the corresponding HTML header tags are generated in the implementation of the above-mentioned methods.

```
1 «DEFINE AIU_Rules FOR AIU»
2     //Main Information Section
3     $Zone = $Page->AddInformationZone(«ManagerClass.Header.Level»,"«ManagerClass.Class.id»");
4     «EXPAND NavigationalRelationship::NavRelationship_rule(ManagerClass)
5             FOREACH NavigationalRelationship-»
6 «ENDDEFINE»
7
8 «DEFINE NavRelationship_rule (ManagerClass mc,) FOR NavigationalRelationship-»
9     «LET this.Target.ComplementaryClass AS cc -»
10    //Complementary Information Section
11    $Detail«cc.id» = «pZoneId»DetailRelationship(«cc.Header.Level»,«cc.id»);
12 «ENDDEFINE»
```

Figure 12 Code generation rule for structural mark-up.

```
1 «DEFINE Context_Rules FOR NavigationalContext»
2     «FILE "/Pages/" + id + ".php"-»
3 <?php
4     $Page = new Page("«alias»");
5     «IF this.AccesiblePage.Breadcrumb == true»
6         $Page->ShowBreadcrumb();
7     «ENDIF-»
8     «ENDFILE-»
9 «ENDDEFINE»
```

Figure 13 Code generation rule for navigation resources.

To support the navigation resources requirement, a code-generation process must also be redefined. From the modelling perspective, the breadcrumb mechanism can be enabled in any

navigational context using an UML annotation (See Figure 11) for specification. The code generation rule (see Figure 13) is quite straightforward: if the breadcrumb mechanism is set to true (line 5), the breadcrumb is included in the generation of a new page (line 4). Regarding the site map mechanism, the selected approach was to include a site map context in any application generated with OOWS, as this is a common requirement. This last change was included into the PHP framework. Thus, it is not guided by the models and must therefore be disabled programmatically.

Finally, in order to include the good navigation requirement, the purpose of each navigation application must be defined explicitly. Figure 14 shows the rule modification for performing this goal. The operation InternLinkTo (line 5) is responsible for generating an HTML link for each navigational relationship from OOWS models. A second parameter has been included in this method in order to set the title attribute of the HTML link generated.

```
1 «DEFINE NavigationLink_rule (String pHeader, NavigationalClass pRelatedClass)
2     FOR ContextRelationship-»
3     «IF this.PageLink.internal == true»
4             «pHeader»«pRelatedClass.Class.id»_«LinkAttribute.Attribute.ref»
5             ->InternLinkTo("«TargetContext.id»", "«this.PageLink.purpouse»");
6     «ENDIF-»
7 «ENDDEFINE»
```

Figure 14 Code generation rule for good navigation.

5.5 Lessons Learned

The main lesson learned is that following the proposed modifications, navigational accessibility requirements from the 23andMe case study are satisfied. With few modifications at both modelling and code generation phases, accessibility is clearly improved in the context of the OOWS method.

Additionally, in the application of the AWA approach in the extension of the OOWS method, several interesting points should be noted. First, it is interesting that accessibility was not considered when the original method was designed. Hence, a preliminary accessibility analysis of the OOWS-generated applications showed several accessibility barriers. Instead of performing a method-specific analysis about how to solve those issues, the AWA support provides a concise solution and has simplified the process to carry out the required changes.

Of further interest is the application of AWA code patterns in the OOWS code generation process. Using some of the code patterns as the default choice in the code generation process, a high level of accessibility is ensured without analyst intervention. Nevertheless, it is true that in order to achieve the best degree of accessibility, some accessibility properties must be adjusted in the modelling phase. This is because some accessibility requirements are guided by Web application requirements.

Although this work focuses on navigational requirements, other accessibility requirements proposed by the AWA approach were also analysed to be included into the OOWS method, such as XHTML standard support. However, these requirements do not have a specific counterpart at the modelling level and thus only impact the code generation process. Other requirements such as that to make text content readable are not supported in any phase of the method and, therefore, cannot be considered. From this experience, it can be concluded that a small sub-set of accessibility requirements

can be addressed from a full model-driven perspective. Nevertheless, if it is possible to represent accessibility requirements as models applying a model-driven development (MDD) method, this significantly decreases development time.

6 Conclusions

Prior to the publication of this paper, few methodological frameworks considering the inclusion of accessibility in the Web development process could be found. The evaluation of accessibility in final code is not enough, since accessibility requirements must be managed from the design phase in order to ensure their systematic inclusion.

The AWA approach presented in this paper provides a set of accessibility requirements in response to this need. The approach is focused on the inclusion of accessibility requirements defined through the abstraction of WCAG. The requirements are addressed with two accessibility mechanisms that offer the following resources, namely, the AWA_Metamodel and AWA_Code_Patterns, as well as the AWA Integration Strategy. With the application of these accessibility mechanisms in the OOWS method, two principal advantages have been detected, namely, that previously-defined OOWS models can be extended with the accessibility criteria and the process whereby accessible code is generated is simplified as a result of providing modelling entities close to accessibility concepts.

Finally, the authors are currently working on an expansion of the AWA methodological support framework that provides assistance for WAI-ARIA.

Acknowledgements

This study has been developed with the support of the MAVIR Research Network (S2009/TIC-1542 [www.mavir.net/]), MULTIMEDICA PROJECT(tin201020644-c03-01) and the Spanish Ministry of Science and Innovation through the project, PROS-Req TIN2010-19130-C02-02. Co-financing was received from the ERDF.

References

1. Aenor, UNE 139803:2012, Web content accessibility requirements, http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0049614
2. Arrue, M., Vigo, M. and Abascal, J. Supporting the Development of Accessible Web Applications. J. Universal Access in the Information Society 14(16): 2699-2719. 2008.
3. Ceri, S., Matera, M., Rizzo, F., and Demaldé, V. Designing data-intensive web applications for content accessibility using web marts. Commun. ACM 50, 4, 55-61. 2007.
4. Bohman, P. R. and Anderson, S. A conceptual framework for accessibility tools to benefit users with cognitive disabilities. In Proceedings of the 2005 international Cross-Disciplinary Workshop on Web Accessibility (W4a) (Chiba, Japan, May 10 - 10, 2005). W4A '05, vol. 88. ACM, New York, NY, 85-89. 2005.
5. Bundesministerium der Justiz (BMJ). Barrierefreie Informationstechnik-Verordnung. (BITV) 2.0. 2011. http://www.gesetze-im-internet.de/bitv_2_0/index.html

6. CEN/CENELEC/ETSI Joint Working Group M 376. European Accessibility Requirements for Public Procurement of Products and Services in the ICT Domain (European Commission Standardization Mandate M 376, Phase 2), http://www.mandate376.eu/

7. Del Fabro, M., Bézivin, J., Valduriez, P.: Weaving Models with the Eclipse AMW plugin. Eclipse Summit Europe, Esslingen, Germany. 2006.

8. Efftinge, S. and C. Kadura. Xpand Language Reference. 2006. http://www.openarchitectureware.org/pub/documentation/4.0/r20_xPandReference.pdf

9. European Commission, Europe's Digital Agenda, Web Accessibility, http://ec.europa.eu/ipg/standards/accessibility/index_en.htm

10. Fons J., Pelechano V., Pastor O., Valderas P., Torres V. Applying the OOWS Model-Driven Approach for Developing Web Applications. The Internet Movie Database Case Study. Chapter 5 in Web Engineering: Modelling and Implementing Web Applications, Rossi, G. et al. Springer. 2008.

11. Forrester Research, Inc. Aligning Availability and Accessibility with Customer Needs And Desires, March 27, 2009. http://forrester.com

12. Freire, A. P., Goularte, R., and de Mattos Fortes, R. P. Techniques for developing more accessible Web applications: a survey towards a process classification. In Proceedings of the 25th Annual ACM international Conference on Design of Communication (El Paso, Texas, USA, October 22 - 24, 2007). SIGDOC '07. ACM, New York, NY, pp 162-169.2007.

13. Guzman, A. R., V. López, et al. Assessing a Web Engineering Method in Practice: a Preliminary Analysis for Personal Genomics Portals XV Ibero-American Conference on Software Engineering, Buenos Aires, Argentina. 2012.

14. Harper, S. and Yesilada, Y., Web Authoring for Accessibility (WAfA). Web Semantic. 5, 3, pp. 175-179. 2007. http://dx.doi.org/10.1016/j.websem.2007.05.001

15. Henry, Shawn Lawton. Just Ask: Integrating Accessibility Throughout Design. Madison, WI: ET\Lawton. 2007. www.uiAccess.com/justask/

16. INSTAC (Information Technology Research and Standardization Center, JAPAN), JIS X 8341-3 Web Accessibility International Standards Research Working Group http://waic.jp/docs/jis2010-understanding/

17. ISO, ISO/IEC DIS 40500. Information technology: W3C Web Content Accessibility Guidelines (WCAG) 2.0. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=58625

18. Jeschke S., Pfeiffer O., and Vieritz H. Developing Accessible Applications with User-Centered Architecture. International Conference on Computer and Information Science (IEEE), Portland/Oregon, May 2008.

19. Lazar, J., Dudley-Sponaugle, A., Greenidge, K. Improving Web Accessibility: A Study of Webmaster Perceptions. Computers and Human Behavior. Vol. 20 No. 2, pp. 269-288. 2004.

20. Martín, Adriana, Rossi, Gustavo, Cechich, Alejandra and Gordillo, Silvia. Engineering Accessible Web Applications. An Aspect-Oriented Approach. World Wide Web (2010) 13:419–440. 2010.

21. Moreno, L., Martínez, P., Contreras, J. and Benjamins, R. 2005. Towards Accessible Semantic Web Applications, DC - International Conference on Dublin Core and Metadata Aplications, Madrid Spain, pp. 87-95, September 2005.

22. Moreno, L., Martínez, P. and Ruíz-Mezcua, B.: Inclusive Usability Techniques in Requirements Analysis of Accessible Web Applications. WISE 2007. Workshops IWWUA: 423-428, 2007. http://www.springerlink.com/content/7641641778416507/

23. Moreno López, Lourdes. Ph.D These: "AWA, Methodological Framework in the Accessibility Domain for Web Application Development", Advisor: Paloma Martínez Fernández, Universidad Carlos III de Madrid, Computer science department, 2010. http://www.sigaccess.org/community/theses_repository/phd/lourdes_moreno.php

24. Moreno, L., Martínez, P., Ruíz, B. and Iglesias, A.: Toward an Equal Opportunity Web: Applications, Standards, and Tools that Increase Accessibility. IEEE Computer 44(5): 18-26, 2011. http://www.computer.org/portal/web/csdl/doi/10.1109/MC.2010.370

25. Newell, A.F. & Gregor, P. User Sensitive Inclusive Design: in search of a new paradigm. En: CUU 2000 First ACM Conference on Universal Usability, 2000.

26. Nykänen O., Kaikuvuo I., Adapting Web Accessibility Guidelines to an Application Development Process, International Design for All Conference, Rovaniemi, Finland, 2006.

27. Olsen, M.G.: How Accessible is the Public European Web. 2008. http://www.mortengoodwin.net/publicationfiles/how_accessible_is_the_european_web.pdf

28. Ontario- e-Laws. Integrated Accessibility Standards made under the "Ontario Regulation 191/11" (Accessibility for Ontarians With Disabilities Act, 2005), June 7, 2011, Ontario.ca. http://www.e-laws.gov.on.ca/html/source/regs/english/2011/elaws_src_regs_r11191_e.htm

29. Pastor, O., Molina, J.C.2007. : Model-Driven Architecture in Practice. A Software Production Environment Based on Conceptual Modelling. Springer-Verlag, Berlin Heildeberg. 2007.

30. Plessers, P., Casteleyn, S., Yesilada, Y., De Troyer, O., Stevens, R., Harper, S., and Goble, C. 2005. Accessibility: a Web engineering approach. In Proceedings of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, pp. 353-362. 2005.

31. Pressman, R. S. Applying Web Engineering. In: Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill. 2005.

32. Référentiel Général d'Accessibilité pour les Administrations (RGAA), Le portail de la modernisation de l'Etat. http://www.references.modernisation.gouv.fr/rgaa-accessibilite

33. Section 508 of Rehabilitation Act. http://www.section508.gov/

34. Sloan. D, Kelly. B, Heath. A, Petrie. H, Hamilton. F, Phipps. L. Contextual Web Accessibility - Maximizing the Benefit of Accessibility Guidelines. In: W4A: Proceedings of the 2006.

International Cross-Disciplinary Workshop on Web accessibility (W4A), Edinburgh, UK, 23-24 May, ACM Press, New York, NY, USA,. pp. 121–131. 2006.

35. Valverde, F. Design and Implementation of an MDA Environment for Web applications development. Master Thesis. Department of Information Systems and Computation. Valencia, Universitat Politècnica de València. 2007
36. Vigo M. and Brajnik G. Automatic web accessibility metrics: Where we are and where we can go, Interacting with Computers, Volume 23, Issue 2, Pages 137-155. 2011
37. Yesilada,Y., Harper, S., Goble, C. and Stevens. R. Dante annotation and transformation of web pages for visually impaired users. In The Thirteenth International World Wide Web Conference, 2004.
38. Xiong, J. and Winckler, M. An investigation of tool support for accessibility assessment throughout the development process of web sites. Journal of Web Engineering, Vol. 7, No.4, pp. 281-298, Rinton Press. 2008.
39. WebAIM, WAVE. Web accessibility evaluation tool. 2012. http://wave.webaim.org/
40. Wieringa, R.J. Design Science as Nested Problem Solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia. pp. 1 12. ACM. 2009. SBN 978 1 60558 408 9
41. Wieringa, R.J. Relevance and problem choice in design science. In: Global Perspectives on Design Science Research (DESRIST). 5th International Conference, 4 5 June, 2010, St. Gallen. pp. 61 76. LNCS 6105. Springer Verlag. 2010. ISBN 978 3 642 13334 3
42. W3C, Web Accessibility Initiative (WAI). 2011. http://www.w3.org/WAI/
43. W3C, Web Accessibility Initiative (WAI), Techniques for WCAG 2.0. 2012. http://www.w3.org/TR/WCAG20-TECHS/
44. W3C, WAI, Web Accessibility Evaluation Tools: Overview. 2012. http://www.w3.org/WAI/ER/tools/
45. W3C, WAI, Web Content Accessibility Guidelines (WCAG) Overview. 2012. http://www.w3.org/WAI/intro/wcag.php