AO-WAD: A Proposal for Accessible Design within Web Engineering Approaches

Adriana Martin[1,2], Gabriela Miranda[1], Viviana Saldaño[1] and Gabriela Gaetán[1]

[1] Unidad Académica Caleta Olivia, Universidad Nacional de la Patagonia Austral, Argentina
[2] GIISCo, Facultad de Informática, Universidad Nacional del Comahue, Argentina

adrianaelba.martin@gmail.com
{gmiranda / vivianas / ggaetan}@uaco.unpa.edu.ar

Abstract. Mature Web Engineering (WE) approaches provide good resources for developing Web 2.0 applications. The state-of-the-art shows that many of these approaches have evolved for providing support to different issues during the development process of these kind of applications, as business-to-business process modeling, context-awareness, RIAs and live-regions or quality factors for improving users' experience. Focusing on Accessibility, having full support usually means being tightly coupled to host process and models, which prevents conveying this support to other WE approaches. In this paper we introduce our proposal, called Aspect-Oriented Web Accessibility Design (AO-WAD), and generalize its use within WE approaches to provide Accessibility support applying Aspect-Orientation techniques. We embed AO-WAD into OOHDM and UWE methods to propitiate an ease understanding through a motivating example.

Keywords: Web Accessibility, WE Approaches, UI Design, Aspect-Orientation

1 Introduction

There is a spectrum of approaches for the Web domain aiming to cover the whole life cycle of Web application development, as UWE [3], OOHDM [11] or WSDM [2]. Many of these approaches have evolved to accompany the new generation of the Web applications and developers. Every day, a huge range of users expects more and better services from the Web and quality factors, as Accessibility and Usability, contribute deeply to improve the experience to all users. The state-of-the-art shows that there are not many proposals for the early design with Accessibility principles in mind and even fewer proposals, provide conceptual tools to fully support Accessibility nature without losing generality, which is required to migrate to other WE approaches. In general, a proposal for including Accessibility design within systematic and unified Web development works only in association with a host WE approach. Therefore, there is a high dependence between host's process and deliverables and the proposed conceptual tools to support Web Accessibility. The consequences are clear, since failing the design principle "low coupling" hinders embedding and easy connection

with other WE approach. For example, Plessers et al. [10] is a well-known proposal that generates annotations for visually impaired users automatically from explicit conceptual knowledge existing during the WSDM [2] design process. The proposal prioritizes Accessibility support using a rule-based mapping model to drive Accessibility annotations, but by means of WSDM's modeling concepts to which these annotations are tightly bound. On the other hand, Moreno et al. [9] defines several constructs in UML[1] meta-model to support the abstraction of Web Accessibility concepts following the standard WCAG [13][14]. Thus, the proposal can be easily implanted into approaches following the MDA[2] paradigm, but at expense of not fully addressing the non-functional, generic and "crosscutting" features of Accessibility. Our proposal for accessible design, called Aspect-Oriented Web Accessibility Design (AO-WAD) [5][6][8], recommends including Accessibility concerns systematically within methods for Web application development. AO-WAD is born to join OOHDM [11] prioritizing Accessibility at the very beginning of the Web design process. While OOHDM provides the main development framework, Aspect-Orientation provides the proper concepts and techniques for fully addressing Accessibility nature within the framework.

In this paper, we introduce AO-WAD as an example of having complete commitment to Accessibility through Aspect-Orientation techniques without losing generality when developing within WE approaches. Supporting this statement, we develop a motivating example within Object-Oriented Hypermedia Design Method (OOHDM) and UML-based Web Engineering (UWE) [3] as host methods, which are two widespread and mature WE approaches.

The rest of this work is organized as follows: in Section 2 we briefly introduce AO-WAD, while in Section 3 we explain the way our proposal provides Accessibility support to Web development processes. Then, in Section 4 we apply AO-WAD to a motivating example using OOHDM and UWE as hosts WE approaches. In Section 5 we achieve some insights about including Accessibility design within Web developments applying Aspect-Orientation techniques. Finally, in Section 6 we present the conclusions and future work.

2 AO-WAD in a Nutshell

The model we envisage to deal with Accessibility concerns within a WE approach is illustrated in Figure 1 [5]. Step 1 (Figure 1 (1)) manages Web application requirements looking for those that involve Accessibility needs. This is because it is at the user's interface level where Accessibility barriers finally show, so we are particularly interested in discovering Accessibility requirements at the user interface (UI) design. Then, Step 2 (Figure 1 (2)) proposes an early capture of Accessibility concrete concerns by developing two kinds of diagrams: the UID with Accessibility *integration points* [5] and the SIG *template* [5] for WCAG 1.0. Step 3 (Figure 1 (3)) aids designers making decisions through the abstract UI model (Figure 1 (3.1)), and then, at Step 4 (Figure 1 (4)) toward its implementation through the concrete UI

[1] See UML specification at <http://www.omg.org/spec/UML/2.0/>
[2] See MDA at <http://www.omg.org/mda/>

model (Figure 1 (4.1)). Thus, given a user's task, the SIG diagram provides the WCAG 1.0 Accessibility checkpoints that "crosscut" the UI widgets (both, abstract and concrete ones; Figure 1 (3.1) and (4.1) respectively), to help to an accessible user experience. Figure 1 (3) shows that at Step 3, our approach provides a supporting tool to assist developers in the implementation of cases, and on the creation of their corresponding models by using reusable components (for a detailed description of AO-WAD see [5]).

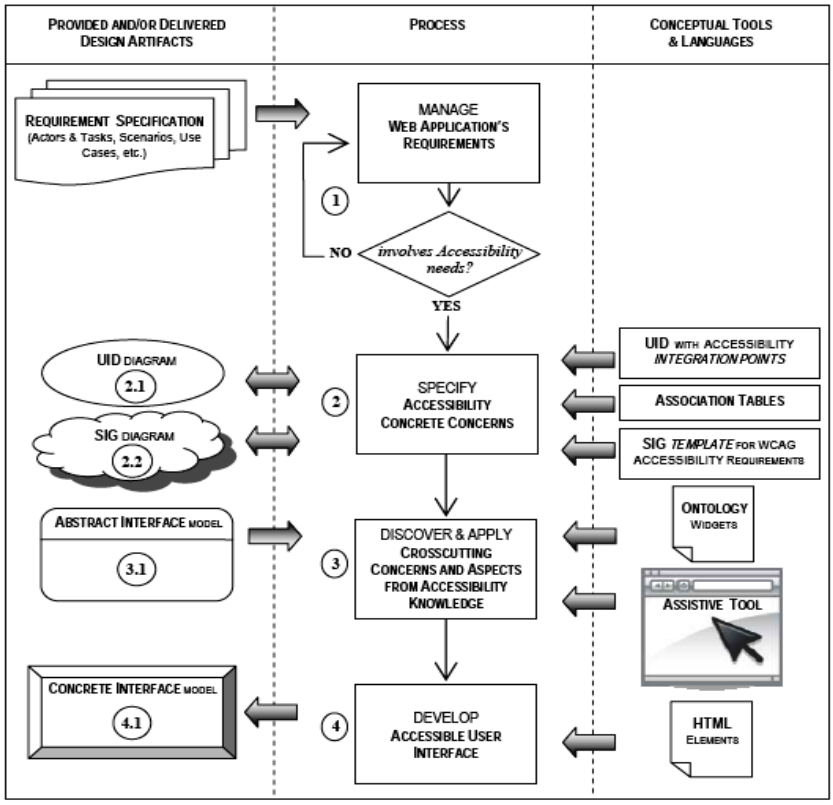

Figure 1. Overview of AO-WAD

In the following section, we show how AO-WAD can be implanted to work not only with OOHDM [11], but also with UML-based Web Engineering (UWE) [3] as one of the most popular and recognized Object-Oriented WE approaches.

3 Systematic Web Development and Accessibility Design

AO-WAD was developed in the spirit of Model-driven paradigm to provide Accessibility support within WE approaches. In Section 2, we describe AO-WAD main process and interaction with OOHDM deliverables to model Accessibility concerns in an Aspect-Oriented manner during Web developments. Figure 2 summarizes the embedding of AO-WAD within OOHDM Model-driven development

process. The UID [12] is the conceptual tool used by OOHDM [11] to state transformations between Web application requirements (Use Case model) and the Conceptual, Navigation and UI models. AO-WAD propitiates the same principle between Web applications requirements and accessible UI models. The interaction between OOHDM models links and reinforces Accessibility needs by applying two conceptual tools: the UID with *integration points* and SIG *template* for Accessibility. The SIG diagram conveys the Accessibility knowledge through WCAG 1.0 *operationalizing softgoals* [5] required to be applied at IU model. Due to Accessibility nature, these Accessibility *softgoals* "crosscut" the UI model more than once causing "crosscutting symptoms". At this point, AO-WAD proposes to address these symptoms by modularizing *softgoals* into Accessibility *aspects*. As Figure 2 shows, the deliverable of the process is an accessible and clean design, which means an OOHDM UI model enriched with Accessibility concerns but free of "crosscutting symptoms".

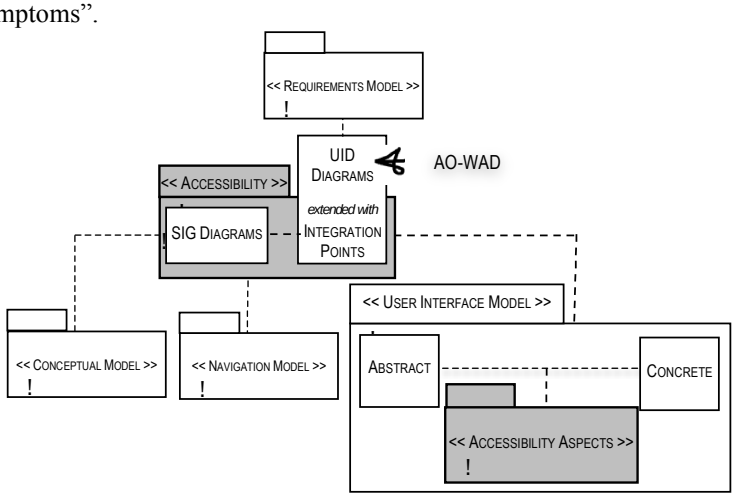

Figure 2: AO-WAD embedded into OOHDM Model-driven Development Process

As another good example of an established WE approach, UWE is based on OMG[3] (modeling and metadata specifications) and uses UML for the analysis and design of Web applications. Figure 3 summarizes the embedding of AO-WAD within UWE Model-driven development process. In UWE [3], the Requirements model consists of two parts: (i) use cases of the Web application and their relationships and, (ii) activities describing use cases in detail. In particular, the Activity diagram is the conceptual tool used by UWE to describe more accurately each use case. UWE uses the Activity diagram to state transformations between Web application requirements and the Content, Navigation and Presentation models. Thus, as Figure 3 shows, AO-WAD embeds into UWE extending the Activity diagrams with *integration points* and through the SIG diagrams convey Accessibility concerns as WCAG 1.0 *operationalizing softgoals*, which "crosscut" the Presentation model causing "crosscutting symptoms". At this point and as we explained before, AO-WAD proposes to address these symptoms by modularizing *softgoals* into Accessibility

[3] See OMG at <http://www.omg.org/technology/documents/modeling_spec_catalog.htm>

aspects. Figure 3 shows the deliverable of the process is an accessible and clean design, which means a UWE Presentation model with Accessibility concerns but free of "crosscutting symptoms".

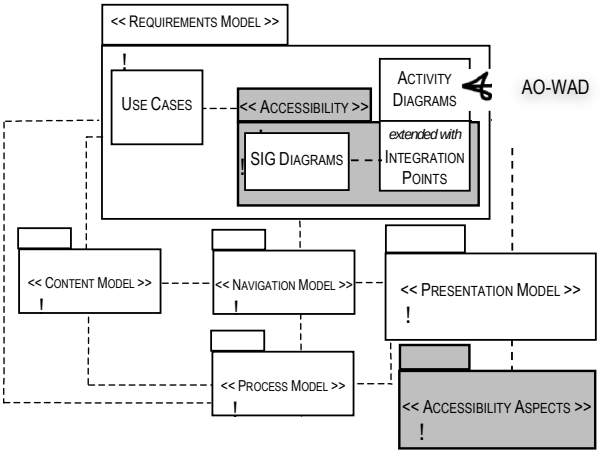

Figure 3: AO-WAD embedded into UWE Model-driven Development Process

In order to ease understanding of AO-WAD within systematic Web development processes, we develop a motivating example in the following section.

4 An Accessible UI for the Students Login

We describe the embedding of AO-WAD within OOHDM and UWE approaches using the following use case specification *"Login a Student given the Student's ID and Password"*:

Use Case: Login a Student given the Student's ID and Password	
Brief Description**:** This use case describes how a Student logs into the SUI Guaraní registration system.	
Success End Condition: The Student is now logged into the system.	
Primary Actor: Student	
Description **Main Success Scenario**:	
Step	Action
1.	The system requests that the Student enter his/her ID and Password.
2.	The Student enters his/her ID and Password.
3.	The system validates the entered ID and Password and logs the Student into the system.
Extensions:	
Step	Branching Action
3.a	The Student enters an invalid ID and/or Password, the system displays an error message, the use case ends.

The use case above describes the Web application's requirements for the student's login and functionality that comprises user-system interaction; as we can see at the first step of the main success scenario, the student is requested by the system to enter his/her ID and Password. Since very often a specification based only on use cases is not enough [12], different kinds of refinement techniques are used to obtain a more detailed specification of functional requirements. OOHDM applies UID technique [12] to model user-system interactions and to specify the information that requires input from the user and choices that allow changes between interactions. On the other hand and following the principle of using UML whenever possible for specification, UWE refine requirements with Activity diagrams for the main stream of the task to be performed. Figure 4, illustrates the UID and Activity diagrams providing a more detailed specification for the login use case in OOHDM and UWE respectively.

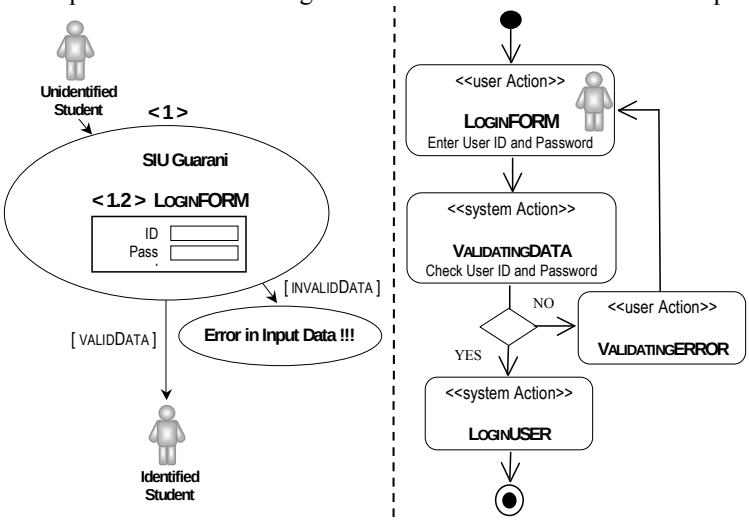

Figure 4: Use Case specifications with UID diagram (left) and Activity diagram (right)

As we already see in Section 2, looking at Step 1, AO-WAD proposes to examine the Web application requirements for the use case above, to identify Accessibility concerns during the user-system interaction. It is clear in this specification that the form element is the key UI element to help achieve an accessible student's login. Following, in Sections 4.1 and 4.2 we focus on modeling issues at Steps 2 (Figure 1 (2)) and 3 (Figure 1 (3)) respectively, as the main steps when implanting AO-WAD within WE approaches.

4.1 Specifying Accessibility Concrete Concerns

When developing with OOHDM, AO-WAD proposes at Step 2.1 extending the UID diagram with *integration points* to supports an early registration of Accessibility concerns. This conceptual tool attaches an Accessibility *integration point* to each one of those UI elements with impact on the dialog required by the use case functionality and modeled by the UID diagram. Looking for the same modeling purpose, AO-WAD Step 2.1 can be also satisfied when developing with UWE, extending the UML

Activity diagram with Accessibility *integration points*. Figure 5 illustrates the UID and UML Activity diagrams enriched with an *integration point* that allows an early record of Accessibility concerns for the UI element --i.e. HTML related controls. Following again the UWE principle of using UML whenever feasible for specification, Figure 5 shows two possible ways of attaching the Accessibility *integration points* to these diagrams: (i) including an UML Note modeling construct or (ii) defining an OCL[4] expression. As we see, integrating Step 2.1 proposed by AO-WAD into the requirement model is straightforward for both WE approaches.

Then, AO-WAD proposes at Step 2.2, the specification of Accessibility *softgoals* through a SIG tree. When developing with OOHDM, the SIG diagram is a consequence of instantiating the SIG *template* taking the UID with *integration points* as input --i.e the early registration of Accessibility concerns for those UI elements, shown by Figure 5 (left), which are core to the required functionality. The SIG diagram specifies Accessibility *operationalizing softgoals* to be satisfied for reaching the WCAG 1.0 level of compliance.

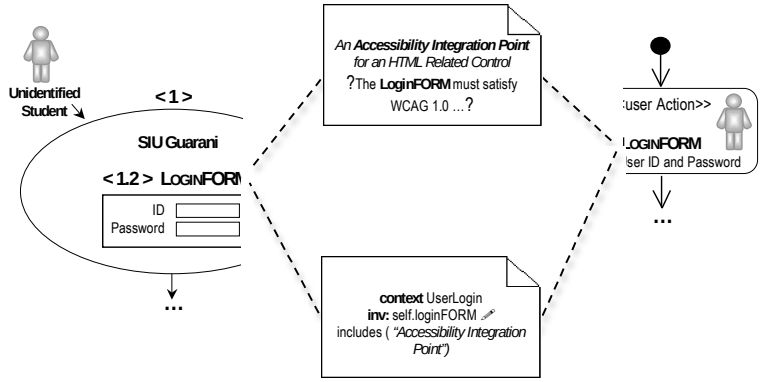

Figure 5: UID diagram (left) and UML activity diagram (right) with Accessibility *integration points*

On the other hand and applying the same modeling purpose, UML Activity diagram extended with *integration points*, shown by Figure 5 (right), provides the required input for developing the SIG diagram within UWE. Although the SIG *template* is not an UML specification tool, it can be easily transformed into a XML tree structure and work with other UML diagrams within the philosophy of the Model-driven paradigm. Therefore, there are no major problems for including Step 2.2 proposed by AO-WAD during the development process of both WE approaches.

4.2 Solving Accessibility Crosscutting Symptoms

AO-WAD proposes at Step 3, the specification of Accessibility *aspects* to avoid "crosscutting symptoms" resulting from applying Accessibility *operationalizing softgoals* to elements comprising the UI model. At the UI modeling stage, OOHDM delivers an Abstract UI model [11] whose vocabulary is established by the Abstract

[4] See OCL specification at <http://www.omg.org/spec/OCL/2.0/>

Widget Ontology extended by AO-WAD [5] to support new elements required by current UI, which are dynamic and with a high degree of complexity. Similarly, UWE delivers a Presentation model [3] from a Meta-model for modeling UI elements.

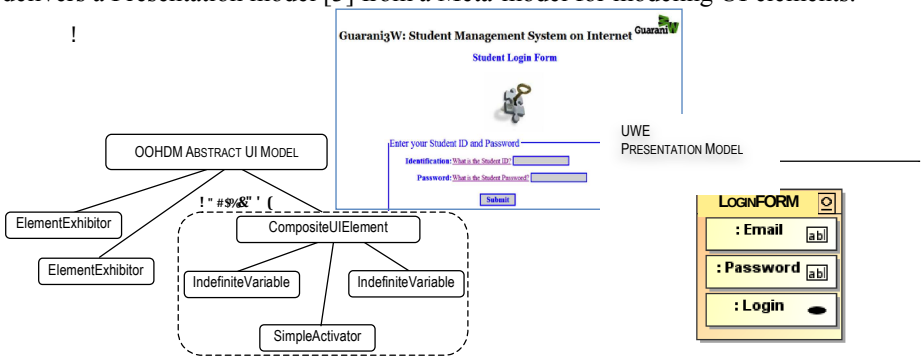

Figure 6: Abstract Interface Model in OOHDM (left) and Presentation Model in UWE (right)

Figure 6 shows the Abstract UI model delivered by OOHDM (left), the Presentation model provided by UWE (right) for the screenshot (top) corresponding to the login example. In first place, AO-WAD recommends discovering "crosscutting symptoms" that manifest when applying Accessibility *operationalizing softgoals* to the UI model --i.e. OOHDM Abstract Interface model and UWE Presentation model.

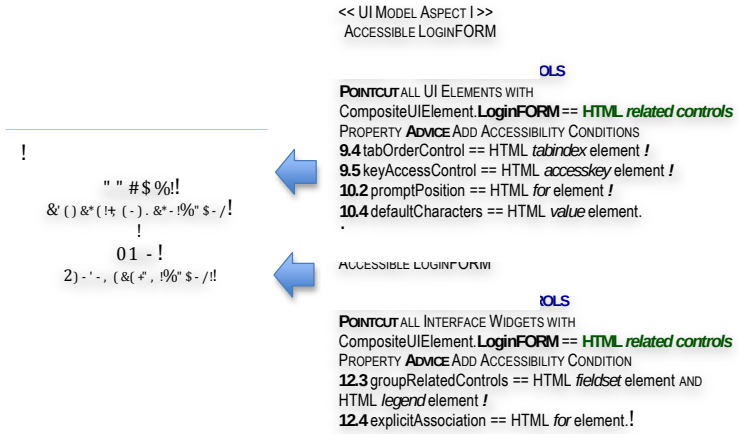

Figure 7: Specification of Accessibility Aspects conveying Accessibility concerns

These *operationalizing softgoals* are spread out and intermixed through the components of the login form UI element, causing "scattering" and "tangling" symptoms. Then, AO-WAD prescribes eliminating these symptoms through a modularization process that applies *aspects* to provide Accessibility support at the user's technology and layout. Thus, *aspects* modularize *operationalizing softgoals* to be satisfied for properly convey the Accessibility concerns required by UI elements. As Figure 7 depicts through a pseudo code, Aspect-Orientation provides a mechanism called "weaving", which requires that each *aspect* must specify "where or how" should be invoked and "what" should be injected into the core --i.e. a concrete UI model.

5 Discussing WE Approaches from the Accessibility Perspective

We have been working for a while on Accessibility [4] and particularly on Accessibility design at early stages of Web applications development [5][6][7][8]. Particularly, we have been applying concepts from Aspect-Orientation in association with the WCAG 1.0 document to deal with Accessibility concerns within WE approaches. In reference to the last statement, we must provide two preliminary clarifications about the rationale behind choosing Aspect-Orientation and the WCAG 1.0 document [13] (instead of the WCAG 2.0 document [14]) for treating Accessibility. In first place, it is not just a coincidence that during this work we refer to Accessibility as "concerns", since the term "concern" from the AOSD perspective describes accurately the non-functional, generic and "crosscutting" features of the Accessibility nature. Second, we based our work on the WCAG 1.0 document, which since 1999 is keeping its value, while the ongoing migration and acceptance process to WCAG 2.0 [14] is completed worldwide. In particularly, Argentina's law number 26.653, called "Guía de Accesibilidad para Sitios Web del Sector Público Nacional", was approved on June 27th 2011 and in August 2011, Argentina became a member of the W3C. Since this law is based on the WCAG 1.0 document gives us extra motivation to continue our work in this version of the WCAG recommendations.

Since the Model-driven paradigm provides a good framework to develop for the Web 2.0, we believe that a proposal to somehow improve the users experience should be able to work within any WE approaches. Although AO-WAD is conceived within OOHDM to fully address Accessibility features, its use can be generalized to work with others, as UWE approach. Extending the Requirement model --e.g. UID diagrams in OOHDM and UML Activity diagrams in UWE, AO-WAD supports an early record of Accessibility concerns embedding smoothly into the main Web development process.

Finally, since AO-WAD is developed to work with the Model-driven paradigm, we would like to highlight advantages/disadvantages of this paradigm and how benefits/affects AO-WAD. On one hand, applying systematic and unified Model-driven approaches brings the benefit of having full documentation and automatic application generation at the expense of introducing some bureaucracy into the development process. Since our proposal suggests the early treatment of the Accessibility concerns through models, we may still be influenced by this reality and its disadvantages --i.e., time and cost consuming, complexity, learning effort, etc. On the other hand, using models and taking advantages of an iterative and incremental development process to deal with Accessibility concerns, allows: (i) going back from UI models to Navigation models to look for alternatives in the navigation path, (ii) assessing the need and relevance of these alternatives to the functionality under develop, and (iii) going forward from Navigation models to UI models to check the Accessibility of the UI related to these alternatives. Thus, the Accessibility of all the alternative navigation paths that may compromise the desired functionality can be evaluated within AO-WAD.

AO-WAD supports accessible Web applications design by embedding Aspect-Orientated techniques into WE developments to proper address Accessibility concerns.

6 Conclusions and Future Work

The application of the Model-driven paradigm to the domain of Web development has resulted in well-known WE approaches, which can be particularly useful because of the continuous evolution of Web 2.0 applications, technologies and platforms. The new generation of Web 2.0 applications must offer user interfaces that enhance the experience and access to all Web users. In this context, we believe that WE approaches provide suitable models to carry with the improvements required by the application under development. In this paper we briefly introduce AO-WAD, which provides complete support to Accessibility concerns by enriching WE models. Following OOHDM and UWE processes, we show that AO-WAD is flexible enough to embed within Web developments. As future work, we will continue working to complete the normalization of AO-WAD and validate its generalized use to systematic developing of accessible Web applications.

References

1. Chung, L., Supakkul, S.: Representing FRs and NFRs: a goal-oriented and use case driven approach. SERA (2004) doi:10.1007/11668855_3
2. Troyer, D., Casteleyn, S., Plessers, P. WSDM.: Web semantics design method. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (eds.) WE, pp. 303–351. Springer (2008)
3. Koch, N., Knapp, A., Zhang, G., Baumeister, H.: UML-based Web Engineering: an approach based on standards. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (eds.) WE, pp. 157–191. Springer (2008)
4. Martín, A., Cechich, A., Rossi, G.: Comparing approaches to Web accessibility assessment. In: Calero, C., Moraga, M.Á., Piattini, M. (eds.) Handbook of research on Web information systems quality, pp. 181–205. Information Science Reference, Hershey (2008)
5. Martín, A., Rossi, G., Cechich, A., and Gordillo, S. Engineering Accessible Web Applications. An Aspect-Oriented Approach. *World Wide Web Journal*, 13(4), 2010, 419-440 doi:10.1007/s11280-010-0091-3
6. Martín, A., Mazalú, R., and Cechich, A. Supporting an Aspect-Oriented Approach to Web Accessibility Design. ICSEA (2010), Francia, doi:10.1109/ICSEA.2010.10
7. Martín A., Cechich, A., and Rossi, G. Accessibility at Early Stages: Insights from the Designer Perspective. W4A (2011), India, doi: 10.1145/1969289.1969302
8. Mazalú, R., Huenuman, F., Martín, A., and Cechich, A. AO -WAD: A Supporting Tool to Aspect-Oriented Web Accessibility Design. ASSE (2011), Argentina.
9. Moreno, L., Martinez, P., Ruiz, B. A MDD Approach for Modeling Web Accessibility. WOST (2008), USA, doi:10.1.1.163.9478
10. Plessers, P., Casteleyn, S., Yesilada, Y., De Troyer, O., Stevens, R., Harper, S., Goble C.: Accessibility: a Web engineering approach. WWW (2005) doi:10.1145/1060745.1060799
11. Rossi, G., Schwabe, D.: Modeling and implementing web applicactions with OOHDM. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (eds.) WE, pp. 109–155. Springer (2008)
12. Vilain, P., Schwabe, D., Sieckenius de Souza, C.: A diagrammatic tool for representing user interaction in UML. UML (2000) doi:10.1007/3-540-40011-7_10
13. W3C: Web Content Accessibility Guidelines 1.0. (WCAG 1.0). http://www.w3.org/TR/WAI-WEBCONTENT/ (1999). Accessed 15 April 2009
14. W3C: Web Content Accessibility Guidelines 2.0 (WCAG 2.0). http://www.w3.org/TR/WCAG20/ (2008). Accessed 25 January 2010.