A Three-Layered Approach to Model Web Accessibility for Blind Users

Adriana Martín[1] Alejandra Cechich[1] Silvia Gordillo[2] Gustavo Rossi[2]

[1]*GIISCo Research Group*
Departamento de Ciencias de la Computación, Universidad Nacional del Comahue,
Buenos Aires 1400, Neuquén, Argentina
Email: {martinae@jetband.com.ar};{acechich@uncoma.edu.ar};

[2]*LIFIA*
Facultad de Informática, Universidad Nacional de La Plata and Conicet
Calle 115 e/ 49 y 50, 1900 La Plata, Argentina
Email: {gordillo@sol.info.unlp.edu.ar};{gustavo@sol.info.unlp.edu.ar}

Abstract

The lack of adequate Accessibility of Web sites remains a problem for blind users and other people with disabilities. The process of engineering accessible Web applications involves several issues, which include content, presentation and pragmatic issues that, fortunately, are currently addressed by some standard models such as the W3C's Accessibility guidelines. In this context, our work proposes an approach to better model Accessibility as a concern specified at different abstraction levels. This paper introduces the approach and illustrates its use through a case study, which includes specific guidelines and techniques.

1. Introduction

Undoubtedly, Web Accessibility is one of the most important aspects upon which the quality of a Web site depends on, since it ensures access by everyone regardless of their permanent or temporary disabilities. In recent years, many research activities have focused on proposing approaches to handle Accessibility from different points of view. Some of these efforts have been oriented to integrate Accessibility into a Web design process, like in [18]. Others focus on evaluation and repair [1,24] or filter and transformation [5,8] of already existing Web site pages. In the context of our research on engineering accessible Web applications, we have developed an evaluation framework, called Web Accessibility Assessment Model (or WAAM for short) [14], to clarify, from an evaluation and classification perspective, the situation at the Accessibility arena. By using this framework we were able to clearly characterize the Accessibility concern from the perspective of a system developer and propose a general solution to incorporate Accessibility into the development life cycle.

Accessibility is a non-functional concern like usability [2], it is generic as it can be instantiated according to the kind of Accessibility requirements and it is crosscutting which means that it has effect on other functional or non functional concerns.

It is well known that crosscutting concerns are particularly harmful because they are *scattered* through all the application components and *tangled* with modules or code which pertains to other concerns. Being able to clearly separate crosscutting concerns early from requirements and through design to implementation and weave them as late as possible allows improving system's modularity and therefore guaranteeing a seamless application's evolution.

These problems have been considered by modern development approaches such as Aspect Oriented Software Development (AOSD); in this paradigm, different concerns are treated as first-class citizen early since requirement elicitation are weaved together using specialized tools (e.g. like compilers). A thorough discussion on these techniques can be found in [3].

Particularly in [4], an approach for modelling and composing navigational concerns shows how the ideas of advanced separation of concerns can be applied both to improve requirement specifications of Web applications and to offer added support for systems evolution. It proposes the early capture of those

0-7695-3008-7/07 $25.00 © 2007 IEEE
DOI 10.1109/LA-Web.2007.9

concerns that affect navigation and their representation using separated analysis and design artefacts. Figure 1 shows an overview of the approach from [4]. The meta-level includes the *system space* and the *meta-concern space*. The *system space* addresses the problem domain description we want to develop, obtained from different sources (interviews, ethnographic studies, analysis of business practices, etc.) and the *meta-concern space* contains a set of typical abstract concerns, which repeatedly manifest themselves in various application domains [4].

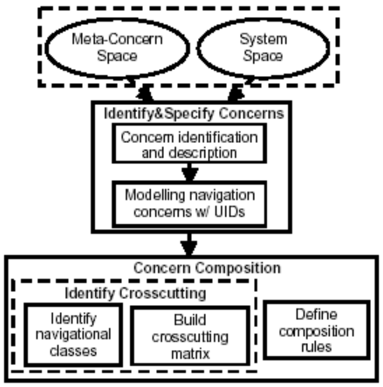

Figure 1: A model for crosscutting navigational concerns

Inspired in this model, we propose a three-layer approach that includes Accessibility as an independent concern. The reason for that is two fold: first we aim to take advantages of the AOSD principles at the early stages of the design process allowing us a deeper Accessibility analysis and treatment; besides taking into account the relevance of Accessibility, we consider that it deserves a specific consideration. This same idea could be also applied of course to other functional or non functional concerns, but this discussion is outside the scope of the paper.

A separate treatment of Accessibility would allow us to better analyse different guidelines and/or recommendations about Accessibility such as the W3C's or Section 508's [20,29], and relate them to the way Accessibility is implemented as a quality attribute of Web interfaces. This analysis should help designers select among different architectural or implementation options.

In addition, using aspect-oriented techniques (such as concern weaving) would facilitate Accessibility propagation and integration by identifying common and variable concerns of an application.

The main contributions of the paper are:

- From a conceptual point of view, a three-layer approach which allows treating different aspects of Accessibility in a systematic way early from requirements.
- From a development point of view, we introduce a set of concepts to incorporate Accessibility issues into User Interaction Diagrams to improve the specification of interaction features.

This rest of the paper is organized as follows. In Section 2, we introduce our proposal for modelling Accessibility using an aspect-oriented view, and we suggest techniques for modelling Web Accessibility. Then, in Section 3 we illustrate our approach through a case study: a student registration system. Conclusions and future work are addressed in the final section of this paper.

2. A Three-Layer Approach

To model Accessibility as a concern, we extended the proposal in [4] as Figure 2 shows. The main driving force of our extension is to specialize the layers to include Accessibility as a specific, separated concern.

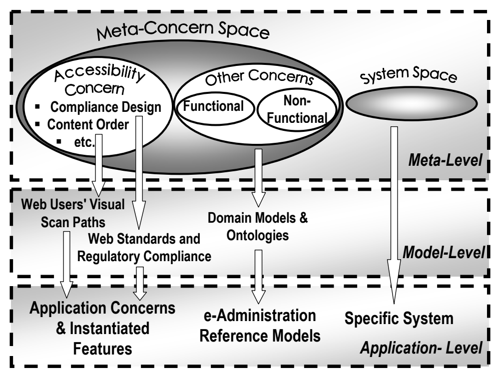

Figure 2: Our model for aspect-based Accessibility

First, our model splits the meta-level into two layers: the meta-level itself and the model-level layers. Basically, the meta-level layer defines concerns that appear several times during the system development. This is the case of the Accessibility concern, which at this level is described in terms of meta-features. The model-level layer instantiates the meta-level layer through appropriate existing models. With the Accessibility concern in mind, these two layers help handle concrete concerns that are relevant to the problem domain description --this description is complemented by the *system space*. Finally, the application-level layer instantiates the model-level layer specifying concrete concerns of an application domain. Let us describe in more detail our three-layer approach:

• ***Meta-Level layer***. As we see in Figure 2, our *meta-concern space* extracts Accessibility as an independent concern from other concerns. Then, our effort is focused on discovering meta-features to better treat, and in consequence, improve Web Accessibility. For example, meta-features for Accessibility can be *compliance design* and *content order*. The first one means conformance to some Web Accessibility design principles articulated by guidelines, regulations, standards or laws; while the second one refers to how to organize de Web pages content based on research reports and studies like quality in use surveys, conducted experiences, patterns catalogues, etc. Additionally, other functional and non-functional recurrent concerns join the *meta-concern space* along with the Accessibility concern.

• ***Model-Level layer***. As we can see from Figure 2, the model-level layer instantiates the Accessibility meta-features into appropriate existing models. Web Standards and Regulatory Compliance, like [6,7,9,10,11,16,20,22,29], can be used to instantiate the c*ompliance design* feature. While Web users' conducted experiences and visual scan studies, like [15,21] and/or Web content structures studies, like [17,27] can be used to instantiate the *content order* feature. In addition, domain models and ontologies may assist in the identification of other concrete concerns pertinent to the problem domain description under development.

The AOSD paradigm provides significant advantages for domain analysis and modelling. For example, in [19] there are several examples where aspects are first-class problem domain concepts that crosscut other problem domain concepts. This work demonstrates that functional and non-functional aspects represent important stakeholder's concerns at the domain level and therefore need a first-order representation.

• ***Application-Level layer***. Now, considering the problem domain description --depicted in the *system space*, the application-level layer instantiates: (i) the model-level layer by extending the Accessibility concern with application concerns and instantiated features, and (ii) the domain models and ontologies by using particular models such as the e-Administration Reference Model [12] for e-learning domains. Concrete concerns are modelled by specifying the interactions related to each one of them. Those concerns involving user interactions are modelled by using the User Interaction Diagram (UID) technique [28]. Accessibility is one of the concerns that are not directly related to the user's semantic dialog, but can influence drastically the application's context of use. Therefore, Accessibility should be analysed as a user interface concern as well as an architectural one, particularly from the quality specification point of view. Then, by considering Accessibility as a non-functional requirement, we extend its analysis with the traditional perspective given by frameworks like [2].

Particularly, a framework for integrating non-functional requirements (NFRs) with functional ones in the use case model is proposed in [23]. In this framework, NFRs are represented as "softgoals" to be "satisfied". To determine satisficeability, design alternatives or decisions (called operationalizing softgoals) are considered; design tradeoffs are analysed; design rationale is recorded and design choices are made. The entire process is recorded in a "Softgoal Interdependency Graph" (SIG) and then the selected design decisions (operationalizing softgoals) can be used as a framework for architecture and design [23]. With this work in mind, we introduce the concept of UIDs's integration points to model the Accessibility concern. The following section further discusses this.

2.1 Accessibility through UIDs integration points

In [2], Chung et al introduced the concept of "association points" to model non-functional requirements along with use cases. By taking a similar perspective, we introduce here the concept of "integration points" to model the Accessibility concern of a user-system interaction at the application-level layer. Particularly, we define two kinds of UIDs integration points:

• *User-UID Interaction* (U-UI) *integration point.* This is an integration point for Accessibility at UID interaction level --i.e. to propitiate an accessible communication and information exchange between the user and a particular interaction of a UID interaction diagram.

• *User-UID Interaction's component* (U-UIc) *integration point.* This is an integration point for Accessibility at UID interaction's component level --i.e. to propitiate an accessible communication and information exchange between the user and a particular UID interaction's component of an UID interaction.

These integration points with different granularity provide two scopes for evaluating Accessibility during the interaction between the user and the system while executing a task. This is especially useful since our approach includes meta-features that can be instantiated with different models to treat the

Accessibility concern. Then, choosing the appropriate granularity and selecting a U-UI or U-UIc integration point allow a better mapping of the elements of those models.

After choosing the Accessibility integration points we propose to develop a SIG for each one. To do that, we take into account concepts from the user interface design literature as follows. The user interface design decision framework in [13] defines the following five topics: (i) structural --specifying the structure of the end users' conceptual model; (ii) functional --specifying functions (operations) which the user can apply to the conceptual objects; (iii) dialog --specifying the content and sequence of information exchanged between the user and the application; (iv) presentation --choosing interaction objects which make up the end users' interface; and (v) pragmatic --dealing with issues of gesture, space, and hardware devices. Since the last three topics are related to the user-system interaction and they are directly involved with Web Accessibility, we decided to introduce these concepts to model the application-level layer.

The dialog topic is directly represented by UIDs since they help modelling the content and the sequence of the information exchange between the user and the system during navigation. However, the presentation and pragmatic topics are relevant too. Therefore, we propose considering the three topics when drawing a SIG for Accessibility. Similar to [23], we propose a SIG template where the Accessibility softgoal denoted with the nomenclature Accessibility[UID integration point] is the root of the tree. The kind of the UID integration point is highlighted into the root light cloud and related to a particular UID interaction or UID interaction's component number (see Figure 3).

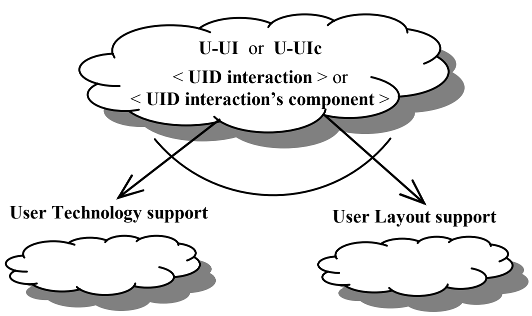

Figure 3: SIG template for Accessibility

From the root node we identify two initial branches: (i) the user technology support, and (ii) the user layout support. The technology support represents the influence of pragmatic and technological topics on the Accessibility softgoal, while the user layout support represents the influence of choosing specific content and objects for user's interaction.

To clarify our proposal, in the next section we illustrated the application-level layer model by using as case study: the Siu Guaraní student registration system.

3. A Case Study

The Siu Guaraní student registration system is been used by a number of public Argentine universities. It offers online information and/or diverse registration functionalities to their students. Since these kind of online systems give support to an educational organization, Accessibility has a main role for students with disabilities --e.g. blinds. In this Section we use the online exam registration function of the Siu Guaraní system to develop our model for integrating Accessibility at the application-level layer.

To make our discussion understandable we present the analysis following the three-level layer in corresponding sub-sections. For the sake of this case, we firstly select the compliance design feature of the Accessibility concern (Meta-Level in Figure 2); and we instantiate this meta-feature with the W3C's WCAG 1.0 guidelines [29] (Model-Level in Figure 2). As we emphasize Model issues we avoid further discussion on the meta-model layer.

3.1 Model Layer Issues

The WACG documents explain how to make Web content accessible to people with disabilities. WCAG is written for content developers as well as for the following [25]: (i) authoring-tool developers to create tools that generates accessible content, (ii) user agent developers to create tools that render accessible content, and (iii) evaluation tool developers to create tools that identify Accessibility issues in content.

Particularly interesting for us are the guidelines 10, 12 and 1. The first two guidelines are related to data-entry forms, which are commonly used to identify users during a registration process. The guidelines 10 and 12 address Accessibility of electronic forms directly; that is, they address the question of what to do to make forms accessible [25]. From guideline 10, we specifically take into account the checkpoints 10.2 and 10.4. The checkpoint 10.2 establishes that "until user agents support explicit associations between labels and form control, for all form control with implicitly associated labels, ensure that the label is property positioned. [Priority 2]"[29]. The checkpoint 10.2 establishes that "until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas. [Priority 3]"[29]. We have to keep in mind that the term "user agent" is used by the W3C as a generic description for any software that retrieves and renders Web content for users, like browsers, mobile phones, screen readers, etc. On the other hand, the term "until user agent" is used by W3C referring to "user agents" that require developers to provide additional support for Accessibility.

From guideline 12, we specifically consider the checkpoints 12.4 and 12.3. The checkpoint 12.4 establishes "associate labels explicitly with their controls. [Priority 2]"[29]. While, checkpoint 12.3 establishes that "divide large blocks of information into more manageable groups where natural and appropriated. [Priority 2]"[29].

It is surprising that all these checkpoints are at most Priority 2 checkpoints, since the issue raised here are critical for a person with a screen reader trying to deal with online forms. To clarify the explanation we introduce some explanation about WCAG's priorities. Each checkpoint has a priority level assigned by the Working Group based on the checkpoint's impact on Accessibility, as follow [29]:

[Priority 1] A Web page Developer *must satisfy* this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use Web documents.

[Priority 2] A Web content developer *should satisfy* this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.

[Priority 3] A Web content developer *may address* this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to Web documents.

Finally, guideline 1 highlights the importance of providing equivalent alternatives to auditory and visual content. This is also a basic functionality for online registration systems since they need to provide option selection menus to the user. It is very common that, to supply these options, Web pages use image map hotspots. From guideline 1 we specifically take into account the checkpoints 1.1 and 1.5. The checkpoint 1.1 establishes that "provide a text equivalent for every non-text element… This includes: images, graphical representations of text, image map regions…, etc. [Priority 1]"[29]. While, the checkpoint 1.5 establishes that "until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of client-side image map. [Priority 1]"[29]. These checkpoints are very clear about including meaningful link text for every image link or image map hotspots to ensure mayor impact on Web page Accessibility.

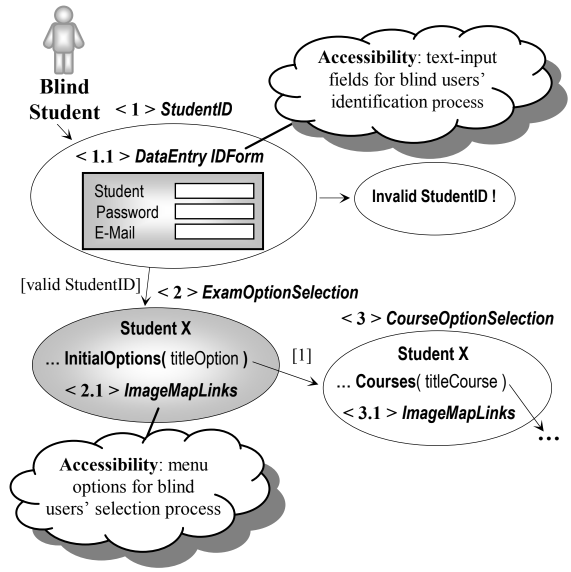

Figure 4: UID with Accessibility integration points

3.2. Application Level Issues

Figure 4 shows a part of the UID for the online exam registration function where a blind student interacts with the Siu Guaraní system. As we can see in the example, we define Accessibility integration points at interactions <1> and <2> representing the *student identification* and the *exam option selection* user-system interaction respectively. More specifically, we define the Accessibility softgoal for the interaction's component <1.1>DataEntryIDForm to ensure accessible text input fields for blind users, and we define two *User-UID Interaction's component* (U-UIc) integration points.

The first one is an identification process. The second one is the Accessibility softgoal for the interaction's component <2.1>ImageMapLink, which ensures accessible menu options for the blind users' selection process.

Then, we apply our SIG template for the two Accessibility softgoals. Figure 5 shows the SIG tree for the Accessibility[BlindStudent-StudentID] softgoal; where a U-UIc integration point for <1.1>DataEntryIDForm interaction's component is highlighted into the root light cloud. Then the *user technology support* and the *user layout support* branches are refined into light clouds --to represent the refined Accessibility softgoal, or dark clouds --to represent operationalizing softgoals. As we said before, we have applied the WCAG 1.0 guidelines related to <1.1>DataEntryIDForm interaction's component. Firstly, looking at the user technology support branch, a distinction between "user agents" and "until user agents" is made in concordance with the distinction made by the W3C's UAAG 1.0 [26]. This is the reason for an or-decomposition of the user technology support branch. If an "until user agent" is the case, satisficing goals related to guideline 10 for checkpoints 10.2 and 10.4 compliance are required. Now looking at the user layout support, satisficing goals related to guideline 12 for checkpoints 12.3 and 12.4 compliance are required.

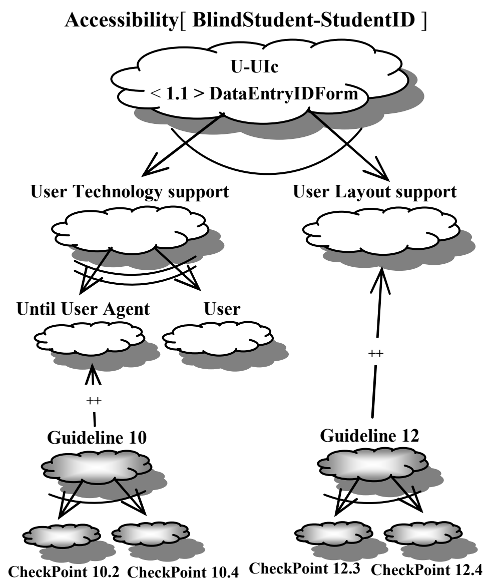

Figure 5: SIG for Accessibility [BlindStudent-StudentID] softgoal

Figure 6 shows the SIG tree for the Accessibility[BlindStudent-ExamRegistration] softgoal. Here, a U-UIc integration point for <2.1> ImageMapLink interaction's component is highlighted into the root light cloud. For brevity reasons we don't give details of this SIG since its analysis can be made similarly as the former.

Since Figure 4 shows only U-UIc integration points, let us briefly discuss the use of U-UI integration points. In our example we use only U-UIc integration points because of the instantiation of the *compliance design* meta-feature with the W3C's WCAG 1.0 guidelines [29]. As we can see, each guideline applies over a particular interaction object of the presentation --our interactions' components. The U-UI integration point would be more useful for a different mapping of the meta-features.

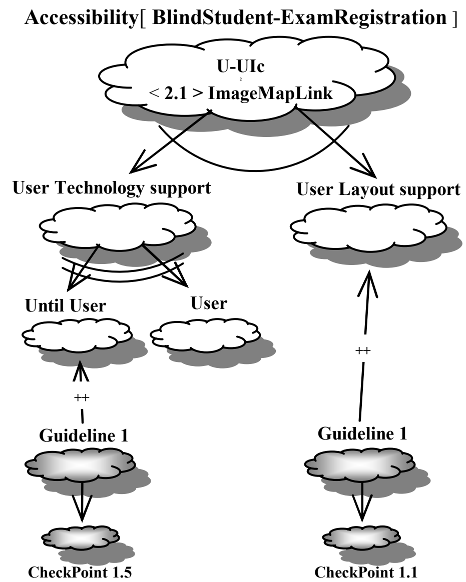

Figure 6: SIG for Accessibility [BlindStudent-ExamRegistration] softgoal

The U-UI integration points treat an UID interaction as a whole, so it is possible to address Accessibility from different points of view considering the organization of the interactions' components, or the interaction's intent for the task described by the UID. For example, we could instantiate the *content order* meta-feature with some Web content structures study, like the Information Structure Taxonomy [17]. This study presents a taxonomy of information structure by user task for voice navigation of Web-spaces. It describes high-level user tasks (Situate, Navigate, Query, Details-on-demand) and information structures (Regions, Menu/Lists, Text Areas, Repeated/Structure Information), which comprise the axes of the taxonomy and show how voice interfaces can support these task and structures. Since this study is focused on how to improve the information structure for voice navigation based on user tasks, U-UI integration points will be the proper choice to evaluate Accessibility.

3.3 Discussion

One of the main advantages of our three-layer approach is the possibility of applying AOSD principles and Accessibility as an independent concern at the early stages of the design. As a consequence, the model propitiates a better analysis of Accessibility since it proposes a fine-grained treatment over this quality factor indeed related to Web applications success. With a fine-grained treatment, designers have the freedom to choose from a wide range of alternatives of implementation to instantiate the Accessibility concern's meta-features. The meta-level and model-level layers prepare the field to carry out smoothly the Accessibility concern at the application-level layer.

At this level, the Accessibility concern is integrated to the user-system interaction. To do this, we propose the UID extended by the integration point technique. In spite of our technique is inspired by the NFR association points [23], substantial differences can be established between them. While the work in [23] associates NFRs at a certain point of the Use-Case diagram, our UID offers a detailed view of the user-system interaction and, a less coarse-grained granularity for defining the Accessibility models integration points. To make the visualization of this fact easier, Figure 7 illustrates how the case study will look like with the NFR association points for Accessibility [23] (see also Figure 4). As in any other design process that uses the Use-Case and UID techniques, these two diagrams can complement each other.

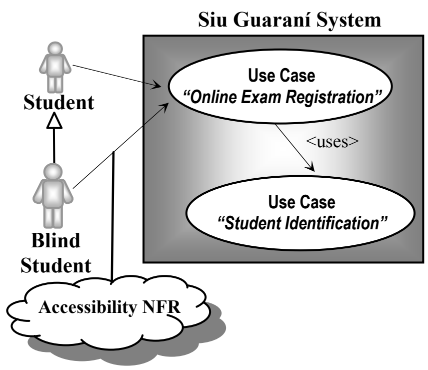

Figure 7: NFR Association points for the Accessibility case study

Another facet to underline our proposal is the fact that introducing concepts from the user interface design literature like [13] to model the application-level layer, would facilitate moving this model to an specific user interface architecture --e.g Model-View Controller (MVC) or Client-Server.

Finally, our model provides extra advantages related to the Accessibility softgoals' SIGs, since this technique provides the basis for Accessibility measurement as a facet of Web quality-in-use. More specifically, our proposal might support the use of qualitative measures to evaluate the Accessibility of a product based on contributions from components of a specific architecture.

4. Conclusions and future work

In May 2006 foreword by Molly Holzschlag said [25]:

"…Berners-Lee's vision has always had to do with the human side of the Web. After all, it's not machines that use the Web, but people… Accessibility is not about disabilities; rather, it's about people getting to shared information that the vision of the Web has made manifest…"

Certainly, saying that Accessibility is a main topic in Web design upon which the success of a Web site depends, has become a landmark statement.

Mostly because of the non-functional, generic and crosscutting nature of Accessibility, our proposal aims at improving Web Accessibility modelling by incorporating elements from the Aspect-Oriented Software Development (AOSD) paradigm. With this in mind, our work proposes a three-layer approach that includes Accessibility as an independent concern. The reason for that is to take advantages of the AOSD principles at the early stages of the design process allowing us a deeper Accessibility analysis and treatment.

At meta-level layer Accessibility concern is defined by meta-features like *compliance design* and *content order*. In this way, our approach lets the freedom of choosing from a wide range of alternative Accessibility models for instantiating at the model-level layer these meta-features. Finally, at the application-level layer we propose integrating Accessibility using UID integration points and then developing Accessibility softgoal' SIG for each one to facilitate evaluation.

However, our model still needs to be improved to include AOSD composition and crosscutting concepts. Also, and as we said before, we can think about using a quality measurement to calculate Accessibility metrics for architectures of Web applications or to support the development of those applications with the Accessibility softgoal in mind.

References

[1] C.Benavídez, J. Fuertes, E. Gutiérrez, & L. Martínez. Semi-Automatic Evaluation of Web Accessibility with HERA 2.0. *In K. Miessenberg et al. (Eds.), 10th International Conference on Computers Helping People with Special Needs (ICCHP 2006)*, pp. 199-106, Springer-Verlag 2006, ISBN 3-540-36020-4, 2006.

[2] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. Non-Functional Requirements in Software Engineering. *Kluwer Academic Publishers*, 2000.

[3] Filman, R., Elrad, T., Clarke, S., Aksit, M. (eds.). Aspect Oriented Software Development. Addison-Wesley, 2004.

[4] S. Gordillo, G. Rossi, J. Araújo, A. Moreira, C. Vairetti, and M. Urbieta. "Modeling and Composing Navigational Concerns in Web Applications. Requirements and Design

Issues". *Proceedings of the 4th Latin American Web Congress (LA-WEB'06)*, pp. 25-31, 2006.

[5] S. Gupta, & G. Kaiser. Extracting Content from Accessibility Web Pages. *International Cross-Disciplinary Workshop on World Wide Web Accessibility (W4A2005)*, pp. 26-30, ACM 2005, ISBN 1-59593-219-4, 2005.

[6] HKSAR. (2001). Hong Kong Special Administrative Region: Digital 21 Strategy. Tips for Improving Accessibility of Web Pages, 2001, from http://www.info.gov.hk/digital21/eng/knowledge/access_main.html.

[7] HREOC. (2003). Human Rights and Equal Opportunity Commission. The Guide to Minimum Website Standards Accessibility, 2003, from http://www.agimo.gov.au/practice/mws/Accessibility/

[8] G. Iaccarino, D. Malandrino, & V. Scarano. Personalizable Edge Services for Web Accessibility. *International Cross-Disciplinary Workshop on World Wide Web Accessibility (W4A2006),* pp. 23-32, ACM 2006, ISBN 1-59593-281-X, 2006.

[9] ISO/DIS 9241-11 (1994) - International Organization for Standardization/Draft International Standard. Ergonomic requirements for office work with visual display terminals (VDTs). Part 11: Guidance on Usability, 1994.

[10] ISO/IEC 9126-4 (2005) - International Organization for Standardization/International Electrotechnical Commission. Software Engineering - Software product quality. Part 4: Quality in use metrics, 2005.

[11] ISO/TS 16071 (2002) - International Organization for Standardization/Technical Specification. Ergonomics of human-system interaction - Guidance on Accessibility for human-computer interfaces, 2002.

[12] R. Koper. Learning Technologies in eLearning: An Integrated Domain Model. In W. Jochems, J. van Merrienboer, E.J.R. Koper, Integrated eLearning, pp. 64-79, London: RoutledgeFalmer, 2003.

[13] J. Larson. Interactive Software: Tools for Building Interactive User Interfaces. *Yourdon Press Computing Series, Prentice Hall,* ISBN: 0-13-924044-6, 1992.

[14] A. Martín, A. Cechich, & G. Rossi. Comparing Approaches to Web Accessibility Assessment. *In Handbook of Research on Web Information Systems Quality*, Idea Group Inc., C. Calero, Mª Á. Moraga, M. Piattini (Eds.), ISSN: 0950-5849 (accepted for publication in 2007).

[15] J. Nielsen, and K. Pernice. Eyetracking Web Usability (Voices That Matter). Publish by New Riders Press, ISBN-10: 0321498364/ ISBN-13: 978-0321498366nResearch on Alertbox Columns, 2007.

[16] PAS 78 (2006). Publicly Available Specification: A Guide to Good Practice in Commissioning Accessible Websites, ICS 35.240.30. Disability Rights Commission (DRC), 2006, from http://www.drc.org.uk/library/website_Accessibility_guidance/pas_78.aspx

[17] M. Pérez-Quiñones, R. Capra, and Z. Shao. The Ears Have It: A task by Information Structure Taxonomy for Voice Access to Web Pages. *Human-Computer Interaction (INTERACT2003),* M. Rauterberg et al. (Eds.) Published by IOS Press, (c) IFIP, pp. 856-859, 2003.

[18] P. Plessers , S. Casteleyn, Y. Yesilada, O. De Troyer, R. Stevens, S. Harper & C. Goble. "Accessibility: A Web Engineering Approach". *Proceedings of the 14th International World Wide Web Conference (WWW2005)*, 2005.

[19] A. Rashid and A. Moreira. Domain Models are NOT Aspect Free. O. Nierstrasz et al. (Eds.): MoDELS 2006, LNCS 4199, pp. 155-169, 2006.

[20] Section 508 (2003). US Federal Government: A Quick Reference Guide to Section 508 Resource Documents, 2003, from http://www.Accessibilityforum.org/paper_tool.html

[21] Software Usability Research Lab (SURL). Service Division to the Human-Computer Interaction (HCI) Laboratory at Wichita State University, from http://psychology.wichita.edu/surl/.

[22] Stanca Law (2004). Italian Legislation on Accessibility: Guidelines containing the technical requirements, the definitions of the different Accessibility levels and the technical methodologies for the testing of Web site Accessibility, 2004, from http://www.pubbliaccesso.it/biblioteca/documentazione/guidelines_study/index.htm

[23] S. Supakkul, and L. Chung. Integrating FRs and NFRs: A Use Case and Global Driven Approach. *Proceedings of the 2nd International Conference on Software Engineering Research and Applications, (SERA2004)*, pp. 29-41, 2004.

[24] H. Takagi, Ch. Asakawa, K. Fukuda, & J. Maeda. Accessibility Designer: Visualizing Usability for the Blind. *International ACM Conference on Assistive Technologies (ASSETS2004),* pp. 177-184, ACM 2004, ISBN 1-58113-911-X, 2004.

[25] J. Thatcher, M. Burks, Ch. Heilmann, S. Henry, A. Kirpatrick, P. Lauke, B. Lawson, B. Regan, R. Rutter, M. Urban, and C. Waddell. (2006). *Web Accessibility - Web Standards and Regulatory Compliance.* FriendSoft Press, ISBN: 1590596382.

[26] UUAG 1.0 (2002). User Agent Accessibility Guidelines. World Wide Web Consortium (W3C) Recommendations, 2002, from http://www.w3.org/TR/WAI-USERAGENT/

[27] D. Van Duyne, J. Landay, and J. Hong. The Design of Sites - Patterns, Principles, and Processes for crafting a Customer-Centered Web Experience. Publish by Addison-Wesley Professional, ISBN-10: 0-201-72149-X/ISBN-13: 978-0-201-72149-2; 2002-2003.

[28] P. Vilain, D. Schwabe and C. de Souza, "A Diagrammatic Tool for Representing User Interaction in UML", *Third International Conference on the Unified Modelling Language*, York, UK, 2000.

[29] WCAG 1.0 (1999). Web Content Accessibility Guidelines 1.0. W3C Recommendations, 1999, from http://www.w3.org/TR/WAI-WEBCONTENT/