Handbook of Research on Web Information Systems Quality

Coral Calero Muñoz
University of Castilla-La Mancha, Spain

Mª Ángeles Moraga
University of Castilla-La Mancha, Spain

Mario Piattini
University of Castilla-La Mancha, Spain

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE

Hershey · New York

Acquisitions Editor: Kristin Klinger
Development Editor: Kristin Roth
Senior Managing Editor: Jennifer Neidig
Managing Editor: Sara Reed
Copy Editor: Lanette Ehrhardt, Susanna Svidunovich
Typesetter: Michael Brehm
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanonline.com

Library of Congress Cataloging-in-Publication Data

Handbook of research on Web information systems quality / Coral Calero, Mª Ángeles Moraga, and Mario Piattini, editors.

p. cm.

Summary: "This book integrates invaluable research on the models, measures, and methodologies of Web information systems, software quality, and Web engineering into one practical guide to Web information systems quality, making this handbook of research an essential addition to all library collections"--Provided by publisher.

Includes bibliographical references and index.

ISBN 978-1-59904-847-5 (hardcover) -- ISBN 978-1-59904-848-2 (ebook)

1. World Wide Web--Handbooks, manuals, etc. 2. Information technology--Handbooks, manuals, etc. 3. Computer software--Quality control--Handbooks, manuals, etc. 4. Application software--Development--Handbooks, manuals, etc. 5. Web services--Handbooks, manuals, etc. I. Calero, Coral, 1968- II. Moraga, Mª Ángeles, 1979- III. Piattini, Mario, 1966-

TK5105.888.H362 2008

004.67'8--dc22

2007032032

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter XI
Comparing Approaches to Web Accessibility Assessment

Adriana Martín
Universidad Nacional del Comahue, Argentina

Alejandra Cechich
Universidad Nacional del Comahue, Argentina

Gustavo Rossi
Universidad Nacional de La Plata and Conicet, Argentina

ABSTRACT

Web accessibility is one facet of Web quality in use, and one of the main actors upon which the success of a Web site depends. In spite of these facts, surveys repeatedly show that the accessibility at the Web for people with disabilities is disappointingly low. At the Web, most pages present many kinds of accessible barriers for people with disabilities. The former scenario encouraged research communities and organizations to develop a large range of approaches to support Web accessibility. Currently, there are so many approaches available that comparisons have emerged to clarify their intent and effectiveness. With this situation in mind, this chapter will discuss the importance of Web accessibility assessment and compare 15 different approaches found in literature. To do so, we provide an evaluation framework, WAAM, and instantiate them by classifying the different proposals. The aim of WAAM is to clarify from an evaluation and classification perspective the situation at the accessibility arena.

INTRODUCTION

The World Wide Web (Web), originally conceived as an environment to allow for sharing of information, has proliferated to different areas like e-commerce, m-commerce, and e-business. Over the last few years, the Web has literally bloomed and the continuous evolution of its purpose has introduced a new era of computing science. A Web application, as any other interactive software system, must to face up to quality properties such as Usability, which ensures the effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments. Particularly, defining methods for ensuring usability and studying its impact on software development is at the present one of the goals that has captured more attention from the research community (Matera, Rizzo, & Toffetti Carughi, 2006; Rafla, Robillard, & Desmarais, 2006). Among these matters, Web accessibility is one facet of Web quality in use, and one of the main actors upon which the success of a Web site depends. An accessible Web site is a site that can be perceived, operated, and understood by individual users despite their congenital or induced disabilities (Irwin & Gerke, 2004; Paciello, 2000). It means having a Web application usable to a wide range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning difficulties, cognitive limitations, limited movement, speech difficulties, photosensitivity and combinations of these. In short, we can say that Accessibility addresses a universal Usability.

Web browsers and multimedia players play a critical role in making Web content accessible to people with disabilities. The features available in Web browsers determine the extent to which users can orient themselves and navigate the structure of Web resources. The notion of travel and mobility on the Web was introduced to improve the accessibility of Web pages for visually impaired and other travelers by drawing an analogy between virtual travel and travel in the physical world (Harper, Goble, & Stevens, 2003). Travel is defined as the confident navigation and orientation with purpose, ease and accuracy navigation within an environment (Yesilada, Harper, Goble, & Stevens, 2004), that is to say, the notion of travel extends navigation and orientation to include environment, mobility and purpose of the journey. Mobility is defined as the easy movement around Web pages supported by visual navigational objects (Yesilada et al., 2004). However, traveling upon the Web is difficult for visually impaired users because the Web pages are designed for visual interaction (Goble, Harper, & Stevens, 2000). Visually impaired users usually use screen readers to access the Web in audio. However, unlike sighted users, screen readers cannot see the implicit structural and navigational knowledge encoded within the visual presentation of Web pages.

Today, many countries are discussing or putting into practice diverse initiatives to promote Web accessibility (HKSAR, 2001; CLF, 2001; European Union, 2002; HREOC, 2003; Cabinet Office, 2003). In spite of these facts, surveys repeatedly show that the accessibility at the Web for people with disabilities is disappointingly low.

The Web Accessibility Initiative (WAI)[1] has developed a set of accessibility guidelines called Web Content Accessibility Guidelines (WCAG 1.0, 1999). The (WCAG 1.0, 1999) recommendations are the established referent for Web accessibility, but there are many other initiatives --e.g. (Section 508, 2003; Stanca Law, 2004; PAS 78, 2006). Table1, borrowed from Loiacono (2004), summarizes a study conducted over 100 American corporations' home pages to specifically examine how well they dealt with the issue of Web accessibility. This study revealed that most of the corporate home pages fail to meet criteria, presenting many kinds of accessible barriers for people with disabilities.

During the last years a large range of approaches have become available to support Web accessibility (Paciello, 2000; Takagi, Asakawa, Fukuda, & Maeda, 2004; Xiaoping, 2004; Yesilada et al., 2004; Plessers, Casteleyn, Yesilada, De Troyer, Stevens, Harper, & Goble, 2005; Leporini, Paternò, & Scorcia, 2006). Tools are useful to assist Web authors at developing accessible content for the Web. Such tools include (Petrie & Weber, 2006): (i) authoring tools that provide guidance on accessibility; (ii) tools that can be used to check for specific accessibility issues, although they were not designed for this purpose; (iii) tools that were developed to visualize specific accessibility issues; (iv) tools that provide easy access to a range of specific checking capabilities; (v) automated evaluation and evaluation and repair tools that evaluate the conformance to some of the standards or guidelines; (vi) testing Web resources with assistive technologies, such as screen readers for blind users and software for dyslexic users, to check how they are rendered in these technologies, and (vii) testing Web resources with disabled Web users to ensure that these groups can easily use the resources. In spite of this diversity, tools for the integration of automatic testing with user and manual testing are still in their initial states of development (Petrie & Weber, 2006). To alleviate these problems, the use of best practices and the application of multiple and different tools must be ensured (Ragin, 2006). However, the heterogeneity of users with different requirements is not yet supported by either automatic tool or tools for manual testing (Benavídez, Fuertes, Gutiérrez, & Martínez, 2006). Actually, there are so many tools currently available that comparisons have emerged to clarify their intent and effectiveness (Brajnik, 2004). Furthermore, a proliferation of organizations is focusing on different aspects of Web accessibility --e.g. WAI, SIDAR[2], CAST[3], AWARE[4], WebAIM[5], ATRC[6], CTIC[7], and so forth.

In this context, this chapter discusses the importance of Web accessibility assessment and compares 15 different approaches found in literature. We provide an evaluation framework and instantiate them by classifying the different proposals. Our accessibility assessment model differentiates three *dimensions*, each one addressing a different concern. The assessment criteria dimension allows distinguishing among the evaluations that can be applied by an approach. While, the assessment deliverables dimension allows categorizing the assessment results characteristics. Finally, the supporting tool dimension considers if the approach counts with specific tool support or not. In Section 3, we describe how to weigh up these concerns when classifying each approach at the resulting grid. In short, the main idea is to make available a method to analyze most relevant aspects of accessibility approaches.

RELATED WORK

There are different approaches to evaluate Web pages accessibility. We discuss some of the most important research works in this area.

Ivory, Mankoff, and Le (2003) have presented a survey of automated evaluation and transformation tools in the context of the user abilities they support. The work discusses the efficacy of a subset of these tools based on empirical studies, along with ways to improve existing tools and future research areas. It aims at evaluating quality of use in three steps: (1) showing a review of automated tools, characterizing the types of users they currently support, (2) given an empirical study of automated evaluation tools showing that the tools themselves are difficult to use and, furthermore, suggesting that the tools did not improve user performance on information-seeking tasks, and (3) describing ways to expand and improve the automated transformation tools in such a way that they make the Web more usable by users with diverse abilities.

Brajnik (2004) has worked on a comparison over a pair of tools that takes into account correctness, completeness and specificity in supporting the task of assessing the conformance of a Web site with respect to established guidelines. The goal of this work is to illustrate a method for comparing different tools that is (1) useful to pinpoint strengths and weakness of tool in terms

Table 1. Relative frequency of barriers/failures on 100 Corporate Home Pages (Loiacono, 2004)

Priority 1: A Web content developer must satisfy this criterion*	Fortune 100 corporate home pages failing to meet criteria
Provide alternate text for all images.	77
Provide alternate text for all image-type buttons in forms.	19
Provide alternate text for all image map hot-spots (AREAs).	17
Give each frame a title.	4
Provide alternate text for each applet.	3
Priority 2: A Web coentent developer should satisfy this criterion.**	
Use relative sizing and positioning (% values) rather than absolute (pixels).	96
Explicitly associate form controls and their labels with the LABEL element.	71
Make sure event handlers do not require the use of a mouse.	63
Use a public text identifier in a DOCTYPE statement.	62
Do not use the same link phrase more than once wehn the links point to different URLs.	46
Do not cause a page to refresh automatically.	8
Create link phrases taht make sense when read out of context.	4
Include a document TITLE	2
Provide a NOFRAMES section when using FRAMEs.	1
Nest headings properly.	1
Avoid scrolling text created with MARQUEE element	0
Priority 3: A Web content developer may address this criterion.***	
Provide a summary for tables.	93
Identify the language of the text.	92
Include default, place-holding characters in edit boxes and text areas.	61
Separate adjacent links with more than white space.	59
Client side image map contains a link not presented elsewhere on this page.	22
Include a document TITLE.	1
Use a public text identifier in a DOCTYPE statement.	1
Section 508	
Provide alternative text for all images.	71
Provide alternate text for all image map hot-spots (AREAs)	26
Explicitly associate form controls and their labels with the LABEL element.	23
Give each frame a title.	10
Provide alternative text for each APPLET	8
Provide alternative text for all image-type button in forms.	8
Include default place-holding characters in edit boxes and text areas	0
Identify the language of the text	0

of their effectiveness, (2) viable in the sense that the method can be applied with limited resources, and (3) repeatable in the sense that independent applications of the method to the same tools should lead to similar results. These properties of the method are partly demonstrated by results derived from a case study using the *Lift* machine and *Bobby* (see these tools in Section3.2).

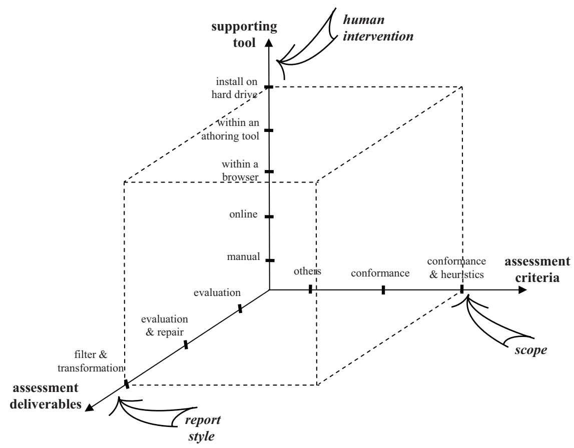
Bohman and Anderson (2005) have developed a conceptual framework, which can be used by tool developers to chart future directions of development of tools to benefit users with cognitive disabilities. The framework includes categories of functional cognitive disabilities, principles of cognitive disability accessibility, units of Web content analysis, aspects of analysis, and realms of responsibility. The authors stated that if tools capable of identifying at least some of the access issues for people with cognitive disabilities are available, developers might be more inclined to design Web pages content accordingly. So, with this vision on mind, the work addresses the next generation of tools with deeper commitment from tool developers to review the underlying structure of the content, the semantic meaning behind it, and the purpose for which it exist: to communicate information to users.

The works cited above agreed on the fact that currently exist fairly abundant accessibility approaches to analyze Web pages' and sites' accessibility. The aim of our work is to provide an accessibility evaluation framework to help clarify the state-of-the-art at the accessibility arena. Differently from the ones cited here, our model—named Web accessibility Assessment Model—is not for classifying approaches from a quality of use perspective (like Ivory's) or from correctness, completeness, and specificity perspective (like Brajnik's), neither it is specifically oriented to tool developers (like Bohman and Anderson's). As an alternative to the former works, we developed a space for comparison, addressing those concerns that we considered most relevant to our purpose of providing a handy accessibility evaluation framework. In this sense, our framework can accomplish and reinforce from an evaluation and classification perspective the former efforts made at the accessibility area.

A SPACE FOR CLASSIFICATION

This section introduces the Web accessibility assessment model (or WAAM for short), a framework for classifying Web accessibility assessment approaches. The organization of WAAM was influenced by our previous work in software quality component models (Cechich & Piattini, 2002), and some related work in quality models for databases (Piattini, Calero, & Genero, 2002) and Web systems (Ruiz, Calero, & Piattini, 2003; Matín-Albo, Bertoa, Calero, Vallecillo, Cechich, & Piattini, 2003). However, we found out that the situation for Web accessibility was fairly more complicated than those cases, because many different dimensions were identified, each one addressing a different concern. Finally, we decided to distinguish between *dimensions* and *factors*. The first ones classify Web accessibility assessment approaches according to the assessment criteria, the assessment deliverables, and the degree of automation through supporting tools. Factors are characteristics that further describe particular aspects of the different dimension's values, such as report style of the assessment

Figure 1. WAAM framework (2007, Adriana Martín. Used with permission.)



deliverables; human intervention required by the supporting tools; and scope of the assessment criteria. WAAM defines three dimensions, which are shown in Figure 1:

- The *assessment criteria* that addresses the way in which the different approaches assess accessibility.
- The *assessment deliverables* that characterizes the results of applying an approach.
- The *supporting tool* that provides space to classify the degree of automation.

WAAM Dimensions

This section covers the three *dimensions* of the framework and their associated features, as depicted in Figure 1. We explicitly detail these *dimensions* showing their respective categories. Also, we explain the *factors* mentioned above and why and how we decided to assign each one to a different *dimension*.

Assessment Criteria

In the analysis we conducted, it was easy to distinguish among approaches applying the rules and guidelines from those applying heuristics or some other evaluations. This is a clear distinction; however, it can cause a great confusion to a potential user of the approaches. As a matter of fact, we discovered that this confusion was present in some cases in which an assessment framework was built upon regulations so that compliance was indirectly reinforced. Therefore, in principle, we decided to clarify this issue by adapting the three assessment criteria proposed by Brajnik (2004). We have differentiated three possible categorizations as follows:

- **Conformance:** It includes the approaches that apply a checklist of principles or guidelines like the ones proposed by (WCAG 1.0, 1999; Section 508, 2003; PAS 78, 2006; Stanca Law, 2004; ISO/TS 16071, 2002).
- **Conformance & Heuristics:** it classifies those approaches that use heuristics in the interpretation and extension of the conformance criteria. These approaches also apply standards but the analysis includes the product's context of use and in some cases other usability properties like user effectiveness, productivity and satisfaction. Examples of this kind of assessment criteria are proposed (ISO/DIS 9241-11, 1994; ISO/IEC 9126-4, 2005; Brajnik, 2006).
- **Others:** These approaches perform evaluations with no direct reference and appliance to accessibility principles and guidelines (WCAG 1.0, 1999; Section 508, 2003; PAS 78, 2006; Stanca Law, 2004; ISO/TS 16071, 2002). It states that the approaches can apply any "other" practice—for example, using an ontology, an heuristic, a markup framework, and so forth, to analyze and treat Web page accessibility and to generate an accessible Web page version.

These three types of *assessment criteria* can produce widely dissimilar results. Most of the available approaches are based on "conformance" criteria, but depending on the applied reference guidelines and in the way they are applied, the results can also be broadly different.

Additionally, as Figure 1 shows, we also took into account a *scope* factor like the one proposed by WebAIM (2006) to examine the *assessment criteria* dimension. In this sense, an approach can be:

- **Simple and limited:** That is, it evaluates just one page at a time and it is often an accessibility evaluation tool available online and as a part of a browser;
- **Very specific:** That is, it focuses on just one element of a Web site, it demonstrates what the site looks like to someone who is blind or has low vision, and it is commonly found in a tool available online and as a part of a browser; or

- **Site-wide:** That is, it examines large sites and checks for a variety of errors, it is an accessibility evaluation tool that usually requires additional software installation.

Assessment Deliverables

In general, there is no consensus on how to define and categorize assessment results characteristics. In our approach, we will follow as much as possible the distinction proposed by the Binghamton University (2001), which identifies three alternative sets: *evaluation*, *evaluation & repair*, and *filter & transformation*. Our value definition is as follows.

- **Evaluation:** These approaches perform a static analysis of Web pages or sites regarding their accessibility, and return a report or a rating.
- **Evaluation & repair:** These approaches perform an evaluation too, but additionally they guide the repairing process by assisting the author in making the pages more accessible.
- **Filter & transformation:** These approaches assist Web users rather than authors to either modify a page or supplement an assisting technology or browser. A filter & transformation process is performed by transcoders that produces a built-in or customized transformed page version. A build-in page version is a consequence of transformations that remove contents or change structure and layout; while a customized page version is a consequence of transformations driven by annotations.

As Figure 1 shows, we also inspect the *assessment deliverables* dimension taking into account the *report style* that the approach produces. In first place, we use here the classification proposed by WebAIM (2006) where the style can be:

- Text-based report, which lists the specific guideline used to scan the page and the instances of each type of accessibility error (some approaches also returns the source code of the Web page where the error occurs);
- Graphic/icon-based report, which uses special icons to highlight accessibility errors and manually checks issues on a Web page (these icons are integrated into the Web page's graphical user interface next to the item on the page with an accessibility issue);
- Evaluation and reporting language (EARL) report, which is a machine readable report; and
- Adaptation-based report. A consideration related to the documents that "filter & transformation" approaches lead us to include this kind of report. As we explained before, a transcoder generates an adapted Web page version and, in general, an intermediate document to drive this Web page adaptation is used during the process. In our framework, these intermediate documents will be considered as adaptation-based reports.

Supporting Tool

This dimension indicates whether the approach has an associated tool support or not. In the former case, this dimension allows a distinction based on where the tool is meant to be available, that is, it functions as a stand along software or embedded into another software or application. On this dimension, we define five supporting tool criteria. Again, we follow the classification proposed by WebAIM (2006) as follows.

- **Manual:** It refers approaches without any supporting tool.
- **Online:** These tools ask the visitor to input the URL of a Web page, choose from a set of evaluation options, and then select a

"Go" button for initializing the accessibility evaluation.

- **Within a browser:** These tools provide accessibility extensions to evaluate the page that is currently at an Internet browser, that is, Internet Explorer, Netscape, Mozilla, and so forth.
- **Within an authoring tool:** These tools function as part of a Web authoring tool, that is, Macromedia Dreamweaver or Microsoft FrontPage, allowing Web developers to examine their content for accessibility in the same environment they are using to create this content.
- **Install on hard drive:** These tools are the most powerful and require their installation on a hard drive or server, like other pieces of software.

As Figure 1 shows, we decided to weight the *supporting tool* criteria with a *human intervention* factor. We apply here a classification that uses the concepts of "automatic test" and "manual test" proposed by Brajnik (2004). An "automatic test" flags only issues that are true problems, while a "manual test" flags issues that are potential problems which cannot be automatically assessed. We named these categories as *none* and *fully,* respectively.

At this point we have to be aware about the limitations of no human intervention. A useful approach must highlight issues that require human evaluation to determine whether they are false positive. That is the reason why we propose another category—*medium*—to represent the case in which the test flags both kind of issues.

Again, some extra considerations related to the human intervention factor for "filter & transformation" approaches are required. In this case, we will use the same classification proposed above for human intervention but with a slightly different implication. Human intervention for "filter & transformation" approaches will refer to the human evaluation needed to mark up issues that require transformations driven by filtering or annotations; while in the former case, human intervention refers to the human evaluation needed to assess issues flagged by the test.

HOW TO USE WAAM

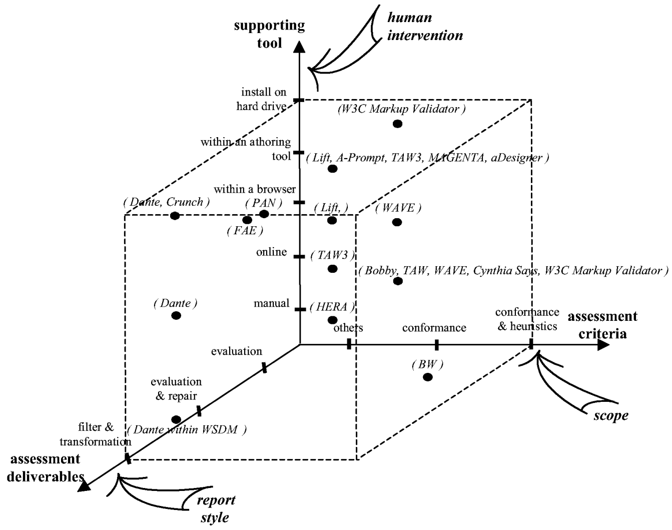
Once the WAAM *dimensions* and *factors* have been defined, this section describes how the WAAM model can be used. Since WAAM defined three dimensions, we will informally refer to the resulting grid as the WAAM *cube.*

Please, note that some approaches have more than one value in each dimension (for instance, there are approaches that can have more than one kind of supporting tool). Thus, we cannot think of the WAAM model as a "taxonomy" for Web accessibility assessments. Rather, each cell in the cube contains a set of approaches: those that are assigned to the cell because the approach applies to the values of the cell's coordinates.

By studying the population of the cube we can easily identify gaps (i.e., empty cells), and also collisions (i.e., overpopulated cells, which means that too many approaches follows similar criteria). Additionally, a given user of the WAAM model who is interested in certain number of characteristics (cells) may quickly obtain the set of approaches that are related to his or her concern.

As mentioned in the introduction, we are currently witnessing a proliferation of approaches for Web accessibility assessment. For the present study, we surveyed the existing literature on these topics, looking for approaches that could provide interesting information for designing and assessing accessibility. For filling the cells of the cube, we iterated over the list of approaches assigning the dimension's values after considering their characteristics. Figure 2 shows the resulting classification, and rationale behind our choices is briefly described below. It is included to clarify the assignment to a particular cell of the cube.

Figure 2. Classification of fifteen web accessibility approaches (2007, Adriana Martín. Used with permission)



Bobby

Bobby has been in the accessibility arena for several years. It was initially developed in 1996 by CAST and it was freely available; now it is operated by WatchFire[8] and has changed their Web site to WebXACT[9]. *Bobby* is an accessibility tool designed to expose possible barriers to Web site use by those with disabilities. *Bobby* checks a Web page and evaluates it to determine how well it addresses the Section 508 standards of the US Rehabilitation Act (Section 508, 2003) and the W3C Web Content Accessibility Guidelines (WCAG 1.0, 1999). The tool can also be configured to complaint one of the three official levels of WCAG 1.0 guidelines (ACompliance—AACompliance—AAACompliance). *Bobby* checks one page at a time and provides a text-based report that lists the specific guideline used to scan the page and the instances of each type of accessibility error. The report highlights issues that are true problems but also issues that are potential problems, so a medium human intervention is required. Due to the reasons explained below we placed *Bobby* in the following cell: *conformance* from the assessment criteria dimension, *online* from the supporting tool dimension and *evaluation* from the assessment deliverables dimension. Finally, we said that *Booby's* test scope is *simple and limited*, *Booby's* human intervention is *medium* and *Booby's* report style is *text-based* report style is text-based.

Lift

Lift was developed by Usablenet[10]. It assists not only to the evaluation process but also the repairing process. *Lift* is an enterprise-wide Web site testing solution that centralizes accessibility and usability management, and like *Bobby*, it allows to test and monitor for compliance with US Section 508 standards (Section 508, 2003), and W3C's Web Content Accessibility Guidelines (WCAG 1.0, 1999). As another similarity with *Bobby*, *Lift* also requires human intervention to check the reported issues. However, there are some differences between the two tools. Firstly, *Lift* generates a variety of Web-based reports to highlight accessibility errors on Web site pages for both executives and individual content creators. Secondly, *Lift* can be a server-based application installed on a hard drive that automatically scans internal and external Web sites. In addition, *Lift* can be an extension to different Web page design applications, for instance, within an authoring tool like Macromedia Dreamweaver and Microsoft Front Page. *Lift* offers an integrated design assistant that guides developers through creating accessible and usable Web pages.

Due to the reasons explained below we placed *Lift* in the cell identified by *conformance* from the assessment criteria dimension, *within an authoring tool* but also *install on hard drive* from the supporting tool dimension and *evaluation & repair* from the assessment deliverables dimension. Finally, we say that *Lift's* test scope is *site-wide oriented*, *Lift's* human intervention is *medium* and *Lift's* report style is *graphic/icon-based*.

A-Prompt

A-Prompt was developed by the University of Toronto at the Adaptive Technology Resource Centre (ATRC). Like *Bobby*, *A-Prompt* evaluates a Web page at a time to identify barriers to accessibility for people with disabilities. But differently from *Bobby* and *Lift*, *A-Prompt* evaluation is aimed to determine the conformance with W3C's Web Content Accessibility Guidelines (WCAG 1.0, 1999) only. Similarly to *Lift*, *A-Prompt* provides the Web author with a fast and easy way to make decisions and to make the necessary repairs. It requires to be installed on hard disk and it runs under Windows 95/98/NT/2000/XP. *A-Prompt* displays a report with dialog boxes and guides the user to fix the problem. Many repetitive tasks are automatically repaired, such as the addition of ALT-text or the replacement of server-side image maps with client-side image maps. Due to the reasons explained bellow, we placed *A-Prompt* in the following cell: *conformance* from the assessment criteria dimension, *install on hard drive* from the supporting tool dimension and *evaluation & repair* from the assessment deliverables dimension. Finally, we say that *A-Prompt's* test scope is *simple and limited*, *A-Prompt's* human intervention is *medium,* and *A-Prompt's* report style is *graphic/icon-based*.

TAW

TAW was developed by the Centre for the Development of Information and Communication Technologies foundation from Spain (CTIC). *TAW* evaluates a Web site to identify accessibility barriers in conformance with W3C's Web Content Accessibility Guidelines (WCAG 1.0, 1999) only. Like *Bobby*, it takes into account the three priorities and the three official levels of WCAG 1.0 guidelines. *TAW* is aimed for Web masters, developers, Web page designers, and so forth. It is a family of free available products[11]: *TAW* online; downloadable *TAW3*; *TAW3 Web Start* and *TAW3 in one click*. The online version has the same properties and functionality as *Bobby*. Downloadable *TAW3* is a desktop application that analyses individual pages or complete Web sites and brings assistance to decision and reparation processes. It has to be installed on a hard disk, it is multiplatform, and it runs over different operating systems like Windows, Mac OS, Unix, and their family, that is, Linux, Solaris, and so forth. Downloadable *TAW3* generates three kinds of report styles of our dimension: text-based, graphic/icon-based and EARL reports. While *TAW3 Java Web Start* has the same functionality as downloadable *TAW3*, its goal is automating the installation process and running a Java-based application with just a click on the Web browser. *TAW3 in one click* is an extension for Firefox browser.

Due to the reasons explained below we have to make a distinction between the *TAW* versions. We place *TAW online* at the same cell as *Bobby,* but in the case of *TAW3*, at the following cell: *conformance* from the assessment criteria dimension, *install on hard drive* and *within a browser* from the supporting tool dimension; and *evaluation & repair* from the assessment deliverables dimension. We say that *TAW online* test scope and the human intervention report style is the same as *Bobby*. Finally, we say that *TAW3's* test scope is *site-wide oriented*, *TAW3's* human intervention is *medium,* and *TAW3's* report style is *text-based, graphic/icon-based* and *EARL*.

HERA

HERA is a multilingual online tool developed by SIDAR Foundation that, like *Bobby* and *TAW,* performs an automatic analysis of as many checkpoints as possible in conformance with W3C's Web Content Accessibility Guidelines (WCAG 1.0, 1999). But, in spite of been an online tool[12], *HERA* supports manual verification and repair assistance, providing extensive help, modified views of the Web page for the evaluation of some checkpoints and storage of evaluation scores and commentaries (Benavídez et al., 2006). *HERA* provides a report generation module that produces two kinds of report styles: text-based and EARL. .*HERA 1.0* was the first version of the tool freely available online to the public in 2003. This version is browser-dependent and uses a set of style sheets written in Cascading Style Sheets (CSS)[13] in order to identify and highlight Web page issues. It allows the evaluator to examine the different issues without having to inspect the source code. *HERA 2.0* is the second version of the tool launched in 2005 to overcome some weaknesses of the previous version. Instead of using CSS to highlight the Web page issues, this new version is browser-independent modified page views because it uses PHP[14] server-side technology.

Due to the reasons explained below, we place *HERA* in the following cell: *conformance* from the assessment criteria dimension, *online* from the supporting tool dimension and *evaluation & repair* from the assessment deliverables dimension. Finally, we say that *HERA's* test scope is *simple and limited*, *HERA's* human intervention is *medium,* and *HERA's* report style is *text-based* and *EARL*.

Dante (Yesilada et al., 2004)

Dante was developed in the Department of Computer Science of the University of Manchester. *Dante* is a semiautomated tool for the support of travel and mobility for visually impaired Web users. The main concept is that travel support could be improved if the objects that support travel are presented in a way that they can fulfill their intended roles and ease the travel. The tool is classified as semiautomatic because the travel analysis is a subjective process, therefore it cannot be fully automated to give as high-quality results as human analysis. That is why a Web page is semiautomatically analyzed an annotated by the tool. *Dante* analyzes Web pages toward semiautomatically: (i) identify travel objects; (ii) discover their roles; (iii) annotate the identified objects by using an ontology; and (iv) transform the Web page with respect to these annotations. To enhance the mobility of visually impaired Web travelers, *Dante* annotates pages with the Web Authoring for Accessibility (WAfA) tool, formerly known as the Travel Ontology (Yesilada et al., 2004), which aims to encapsulate rich structural and navigational knowledge about the travel objects. The tool can be implemented and used on both the server side and the client side. It is more likely that it will be implemented on the client side. In *Dante*, the Mozilla plug-in version of COHSE[15] is used for annotation and the prototype transformation part of *Dante* is also implemented as a plug-in to Mozilla. By using a plug-in approach, the transformer and the annotator can access the DOM[16] object built by the browser and they can base the transformations and annotations on this intermediate document. In (Plessers et al., 2005), the *Dante* annotation process is integrated into the Web Site Design Method (WSDM) that allows Web sites and Web applications to be developed in a systematic way. The annotations are generated from explicit conceptual knowledge captured during the design process by means of WSDM's modeling concepts. These WSDM's modeling concepts used in the different phases are described in WSDM ontology. To generate code that is annotated with concepts from the WAfA ontology, a relationship between the concepts in the WSDM ontology and the WAfA ontology is established. By using these mapping rules, we can establish a transformation process that takes the conceptual design models as input and generates a set of annotations as a consequence. The transformation process consists of two annotation steps: authoring and mobility, which resemble the original annotation process of the Dante approach. The difference is that the authoring annotation in Dante is manual and based on the HTML source code of the Web site.

Due to the reasons explained below, we decided to differentiate between *Dante* and *Dante within WSDM*. We classify *Dante* as *others* from the assessment criteria dimension, *within a browser* but also *install on hard drive* from the supporting tool dimension and *filter & transformation* from the assessment deliverables dimension. We say that *Dante's* test scope is *simple and limited*, *Dante's* human intervention is *medium* and *Dante's* report style is *adaptation-based*. We classify *Dante within WSDM* as *others* from the assessment criteria dimension, *manual* from the supporting tool dimension and *filter & transformation* from the assessment deliverables dimension. We say that *Dante within WSDM's* test scope is *site wide*, *Dante within WSDM's* human intervention is *fully* and *Dante within WSDM's* report style is *adaptation-based*.

PAN

Personalizable accessible navigation (*PAN*) (Iaccarino, Malandrino, & Scarano, 2006) was developed in the Informatics and Applications Department of the University of Salerno. *PAN* is a set of edge services designed to improve Web page accessibility and developed on the top of a programmable intermediary framework, called SISI: Scalable Infrastructure for Edge Services (Colajanni, Grieco, Malandrino, Mazzoni, & Scarano, 2005). The main goal of *PAN* is to provide efficient adaptation services, that is, services that are able to apply different types of on-the-fly transformations on Web pages in order to meet different users' preferences/needs/abilities. To use *PAN's* set of accessibility services, users have to install the SISI framework that is available as raw source code for Unix/Linux platforms and in a precompiled version for Windows. The installation and the deployment of *PAN* are accomplished by simply using the deployment mechanism provided by the SISI framework. The services provided by *PAN* are grouped into four main categories depending on whether they act on text, links, images or other objects on the HTML page—such as pop-up windows—according to the classification implicitly provided by the Web Content Accessibility Guidelines (WCAG 1.0, 1999). The text-based edge services adapt Web pages by taking into account the rules suggested by W3C to improve accessibility and to enhance, in general, the navigation of Web pages and, more specifically of Cascading Style Sheets (CSS) files. The link-based edge services act on links of Web pages in order to make Web pages more readable when users use assistive technologies such as speech synthesizers, screen readers, and so forth. The filter images edge services remove any image embedded in a Web page by replacing it with a link to it. The GIF animated images are also replace with a static one, by showing its first frame. The easy and smooth navigation service removes advertisements, banners, pop-ups in Javascripts and HTML, and so forth. This service also removes useless and redundant code, white spaces, HTML comments, and so forth.

Due to the reasons explained below, we place *PAN* at the following cell: *conformance* from the assessment criteria dimension, *install on hard drive* from the supporting tool dimension and *filter & transformation* from the assessment deliverables dimension. We say that *PAN's* test scope is *site-wide*, *PAN's* human intervention is *none,* and *PAN's* report style is *adaptation-based*.

Barrier Walkthrough

The *BW: Barrier Walkthrough* (Brajnik, 2006) was developed in the Mathematics and Informatics Department of the University of Udine. *BW* is a heuristic walkthrough method based on barriers[17]. This work defines a barrier as any condition that makes it difficult for people to achieve a goal when using the Web site through specified assistive technology. A barrier is a failure mode of the Web site, described in terms of (i) the user category involved, (ii) the type of assistive technology being used, (iii) the goal that is being hindered, (iv) the feature of the pages that raise the barrier, and (v) further effects of the barrier. Barriers to be considered are derived by interpretation of relevant guidelines and principles (WCAG 1.0, 1999; Section 508, 2003; PAS 78, 2006). To apply *BW* a number of different scenarios need to be identified. A scenario is defined by user characteristics, settings, goals, and possibly tasks of users who belong to given categories. At least categories involving blind users of screen readers, low-vision users of screen magnifiers, motor-disable users of normal keyboard or mouse, deaf users, and cognitive disabled users should be considered (Brajnik, 2006). In the *BW* method, user goals and tasks can be defined only referring the site being tested. For a Web application, one should consider some of the possible goals and tasks usually documented in use cases and cross these goals with user categories to obtain the relevant scenarios. For the information of a Web site, a sample of possible information needs can be considered and crossed with user categories. In this way, each user goal/task will be associated to different sets of pages to test, and these will be crossed to user categories (Brajnik, 2006). Evaluators then analyses these pages by investigating the presence of barriers that are relevant to the particular user category involved in the scenario. Cross-checking a barrier to a set of pages in the context of a scenario enables evaluators to understand the impact of this barrier with respect to the user goal and how often that barrier shows up when those users try to achieve the goal (Brajnik, 2006). Finally, using the *BW* evaluator produces a list of problems associated to a barrier in a given scenario, to a severity level, and possibly to performance attributes that are affected, that is, effectiveness, productivity, satisfaction, safety. The *BW* tries to assist the evaluator in filling the gap created by guidelines for conformance testing, because they often are very abstract to be directly applicable to Web sites.

Due to the reasons explained below, we place *BW* at *conformance & heuristics* from the assessment criteria dimension, *manual* from the supporting tool dimension and *evaluation & repair* from the assessment deliverables dimension. We say that *BW's* test scope is *simple and limited*, *BW's* human intervention is *fully,* and *BW's* report style is *text-based*.

WAVE

Web accessibility Versatile Evaluator (*WAVE*) is a free, Web-based tool to help Web developers make their Web content more accessible. *WAVE* was developed by WebAIM in conjunction with the Temple University Institute on Disabilities[18]. *Wave* facilitates evaluation by exposing many kinds of accessibility errors in the content, as well as possible errors, accessibility features, semantic elements, and structural elements. Like *Bobby*, *WAVE* evaluates pages against guidelines (WCAG 1.0, 1999; Section 508, 2003) and displays instances of different types of errors on the page. *WAVE* is an online service but it can be a tool within a browser too. *WAVE* checks one page at a time and provides a graphic/icon-based report and also an EARL report. These reports list the specific guideline being used to scan the page and the instances of each type of accessibility error. Like *Bobby*, the *WAVE* report highlights issues that are true problems but also issues that are potential problems, so a medium human intervention is required.

Due to the reasons explained below we place *WAVE* at *conformance* from the assessment criteria dimension, *online* but also *within a browser* from the supporting tool dimension and *evaluation* from the assessment deliverables dimension. Finally, we say that *WAVE's* test scope is *simple and limited*, *WAVE's* human intervention is *medium,* and *WAVE's* report style is *graphic/icon-based* and also *EARL*.

FAE

Functional accessibility evaluator (FAE) with the *Web accessibility visualization tool* and the *HTML Best Practices* (Rangin, 2006) were developed by the University of Illinois at Urbana/Champaign (UIUC)[19]. The goal of accessibility at UIUC is to make Web resources more functionally accessible to people with disabilities by improving the navigational structure or ability of users to restyle content for their own needs. The tools support developers in using accessible markup by estimating the use of best practices, and help developers visualize the accessibility of their resources. The tools use the following functional accessibility requirements defined in five major topics: (i) navigation and orientation, (ii) text equivalents, (iii) scripting, (iv) styling, and (v) standards. UIUC developed a set of *HTML Best Practices* that translates the requirements of guidelines (Section 508, 2003; WCAG 1.0, 1999) into markup requirements for implementing common Web page features. This translation of requirements into markup requirements is substantially different from conventional assessment tools like *Lift*, since *FAE* works over the *HTML Best Practices* document instead of over the accessibility principles and guidelines. *FAE* provides a means to estimate the functional accessibility of Web resources by analyzing Web pages and estimating their use of best practices. The test results are linked to both the *HTML Best Practices* document and the *Web accessibility Visualization Tool* for Web developers to find out more information about the results. The *Web accessibility Visualization Tool* is a visualization tool that provides graphical views of functional Web accessibility issues based on the *HTML Best Practices*.

Due to the reasons explained below we place *FAE* with the *Web accessibility Visualization Tool* and the *HTML Best Practices* at *others* from the assessment criteria dimension, *within an authoring tool* from the supporting tool dimension and *evaluation & repair* from the assessment deliverables dimension. Finally, we say that *FAE's* test scope is *simple and limited*, *FAE's* human intervention is *medium* and *FAE's* report style is *graphic/icon-based*.

Crunch

Crunch (Gupta & Kaiser, 2005) is a tool for preprocessing inaccessible Web pages to make them more accessible. *Crunch* is developed as a Web proxy usable with essentially all browsers, for the purpose of content extraction (or clutter reduction). It operates sending the Web browser's URL request to the appropriate Web server, and then applying its heuristic filter to the retrieved Web page before returning the content extracted from that Web page to the browser or other HTTP client. The first step in Crunch's analysis of the Web page is to pass it through a conventional HTML parser, which corrects the markup and creates a Document Object Model (DOM) tree. *Crunch's* heuristic manipulates the DOM representation in terms of tree transformation and pruning operations, rather than working with HTML text. This enables *Crunch* to perform its analysis at multiple granularities walking up and down the tree (Gupta & Kaiser, 2005). One of the limitations of the framework is that *Crunch* could potentially remove items from the Web page that the user may be interest in, and may present content that the user is not particularly interested in. *Crunch* partially addresses this problem by offering two ways. The first one is providing an administrative console, whereby an individual user can adjust the "settings" of each heuristic in order to produce what that user deems the "best" results for a given Web page. But, this manual procedure can be tedious and not appropriate for most users. So, the second one, is automating analogous tweaking by employing another set of heuristics to try to determine whether the DOM-pruning collection of heuristics mentioned above are properly narrowing in on the content. In short, *Crunch* is a Web proxy that utilizes a collection of heuristics, essentially tuneable filters operating on the DOM representation of the HTML Web page, with, among other goals, that the resulting Web page be accessible even if the original was not (Gupta & Kaiser, 2005).

Due to the reasons explained below we place *Crunch* at *others* from the assessment criteria dimension, *install on hard drive* from the supporting tool dimension and *filter & transformation* from the assessment deliverables dimension. Finally, we say that *Crunch's* test scope is *simple and limited*, *Crunch's* human intervention is *medium* and *Crunch's* report style is *adaptation-based*. Note that in spite of *Dante* and *Crunch* share the same classification space, there is a difference between them. While the former applies transformations by annotations using an ontology, *Crunch* applies filtering by pruning using a heuristic framework.

MAGENTA

Multi-Analysis of Guidelines by an Enhanced Tool for Accessibility (*MAGENTA*) with *GAL: guidelines abstraction language* and *GE: guideline editor* (Leporini et al., 2006) is an environment for defining, handling, and checking guidelines for the Web. The goal of such an environment is to support developers and evaluators in flexibly handling multiple sets of guidelines, which can be dynamically considered in the evaluation process. The *MAGENTA* tool has been developed with the intent to check whether a Web site is accessible and usable and to provide support to improve it. Currently, *MAGENTA* supports three sets of guidelines for the Web: a set of guidelines for visually-impaired users (Leporini & Paternò, 2004), the guidelines from the (WCAG 1.0, 1999) and the guidelines associated with the (Stanca Law, 2004). The tool is not limited to checking whether the guidelines are supported but, in case of failure, it also provides support for modifying the implementation in order to make the resulting Web site more usable and accessible. *MAGENTA* has been developed considering the limitations of most current tools, in which the guidelines supported are specified in the tool implementation. In this work the aim is to provide a tool independent from the guidelines to check. The solution is based on the definition of a language for specifying guidelines that are stored externally to the tool. As guideline specification, the XML-based *Guideline Abstraction Language (GAL)* is proposed. In order to facilitate this process, a graphical editor has been designed and added to the *MAGENTA* tool, thus enabling people not particularly expert in handling languages such as X/HTML[20] and CSS to specify the desired guidelines. The *Guideline Editor (GE)* has been designed for assisting developers in handling single as well as groups of guidelines. The tool supports new guidelines definition and various types of editing (Leporini et al., 2006).

Due to the reasons explained below we place *MAGENTA* with *GAL* and *GE* at *conformance* from the assessment criteria dimension, *install on hard drive* from the supporting tool dimension and *evaluation & repair* from the assessment deliverables dimension. Finally, we say that *MAGENTA's* test scope is *site-wide*, *MAGENTA's* human intervention is *medium,* and *MAGENTA's* report style is *graphic/icon-based*. Note that in spite of *MAGENTA* share the same classification space with *Lift*, *TAW,* and *A-Prompt*, there is a difference between them. While *MAGENTA* is a tool independent from the guidelines to check, the other tools are predefined to test and monitor for compliance with (WCAG 1.0, 1999) guidelines or with (Section 508, 2003) standards.

Cynthia Says

Cynthia Says was developed by the International Centre for Disability Resources on the Internet (ICDRI)[21] in collaboration with Hi-Software[22]. It is an automated accessibility checker that can be used to test one Web page per minute. It can generate a report based on (Section 508, 2003) standards or on (WCAG 1.0, 1999) checkpoints, with and additional evaluation of the quality of alternative texts. This evaluation looks at the page for some common authoring tool errors or alt text creation errors. *Cynthia Says* is an online automatic tool available only in English that does not check all the checkpoints and provides no support, beyond a checklist for manual evaluation (Benavídez et al., 2006).

Due to the reasons explained below we place *Cynthia Says conformance* from the assessment criteria dimension, *online* from the supporting tool dimension and *evaluation* from the assessment deliverables dimension. Finally, we say that like *Bobby*, *Cynthia Says's* test scope is *simple and limited*, *Cynthia Says's* human intervention is *medium,* and *Cynthia Says's* report style is *text-based*.

aDesigner

aDesigner (Takagi et al., 2004) was developed in collaboration with alphaWorks Services[23] from IBM research and development labs. *aDesigner* is a disability simulator that helps Web designers ensure that their pages are accessible and usable by the visually impaired. Web developers can use *aDesigner* to test the accessibility and usability of Web pages for low-vision and blind people. The tool looks at such elements as the degree of color contrast on the page, the ability of users to change the font size, the appropriateness of alternate text for images, and the availability of links in the page to promote navigability. The tool also checks the page's compliance with accessibility guidelines. The result of this analysis is a report listing the problems that would prevent from accessibility and usability to visually impaired users. In addition, each page is given an overall score. With this information, Web developers get immediate feedback and can address these obstacles before the pages are published. The platform requirements to install *aDesigner* is Windows 2000 or XP operating systems and Internet Explorer 6.0 or above. Once *aDesigner* is installed on hard drive it performs two kinds of accessibility checks. The first one, checks regulations and guidelines such as (Section 508, 2003; WCAG 1.0, 1999) but also checks the Japan Industrial Standard (JIS)[24] and the IBM's checklist[25]. They call such items to be checked as "compliance items." The second one checks usability problems faced by people with visual impairments, going beyond compliance. An author can experience how low vision users see the Web pages by using "low vision simulation" modes, and an author can also understand how blind users listen to and navigate through the pages by using "blind visualization." *aDesigner* aims at providing Web designers an environment to gain experiences in how low-vision people see a Web page, and how blind people access a Web page by using voice browsers. In short, aDesigner is an assistive authoring tool that aims at simulating disabilities to check the pages real usability while authoring.

Due to the reasons explained below we place *aDesigner* at *conformance* from the assessment criteria dimension, *install on hard drive* from the supporting tool dimension and *evaluation & repair* from the assessment deliverables dimension. Finally, we say that *aDesigner's* test scope is *very specific*, *aDesigner's* human intervention is *medium,* and *aDesigner's* report style is *graphic/icon-based*. Note that *aDesigner* share the same classification space with *Lift*, A-*Prompt*, *TAW3,* and *MAGENTA*. However, the difference among these accessibility tools is that *aDesigner* tries not only to evaluate compliance but goes beyond compliance too.

Table 2. The assessment criteria dimension weigh with the scope factor

	Scope factor ✓ simple and limited	*Assessment criteria*						
	~ very specific	others			conformance			confor-mance & heuristics
	× site-wide	Ontology	Markup	Heuristic	WCAG	Section 508	Another standard / specifica-tion	BW over WCAG
Accessibility approach	Bobby				✓	✓		
	Lift				×	×		
	A-Prompt				✓			
	TAW				✓ ×			
	HERA				✓			
	Dante	✓						
	Dante within WSDM	×						
	PAN				×			
	BW							✓
	WAVE				✓	✓		
	FAE							
	Crunch			✓				
	MAGENTA				×		×	
	Cynthia Says				✓	✓		
	aDesigner				~	~	~	
	W3C MVS						✓	

Figure 3. Analysis of the population at the assessment criteria dimension (2007, Adriana Martín. Used with permission.)

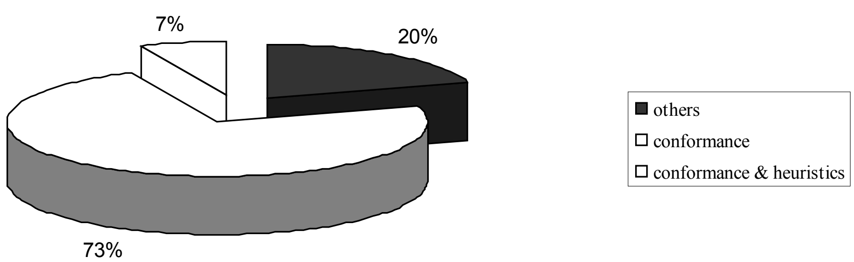


Table 3. The assessment deliverables dimension weigh with the report style factor

Report Style factor		*Assessment deliverables*			
✓	text-based				
~	graphic/icon-based	evaluation	evaluation & repair	filter & transformation	
×	EARL			build-in page version transformations by removing contents or changing structure and layout	customized page version transformations driven by annotations
✧	adaptation-based				
Accessibility approach	Bobby	✓			
	Lift		~		
	A-Prompt		~		
	TAW	✓	✓ ~ ×		
	HERA		✓ ×		
	Dante				✧
	Dante within WSDM				✧
	PAN			✧	
	BW		✓		
	WAVE	~ ×			
	FAE		~		
	Crunch			✧	
	MAGENTA		~		
	Cynthia Says	✓			
	aDesigner		~		
	W3C MVS	✓			

Figure 4. Analysis of the population at the assessment deliverables dimension (2007, Adriana Martín. Used with permission.)

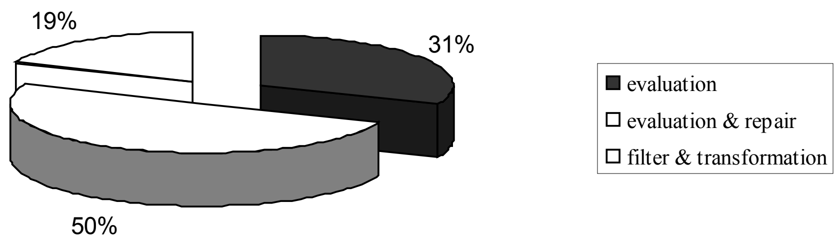
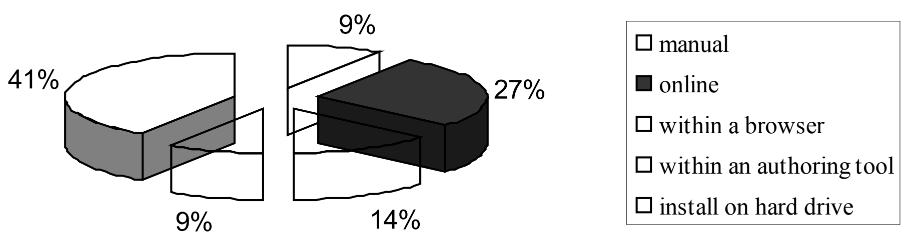


Table 4. The supporting tool dimension weigh with the human intervention factor

Human Intervention factor		*Supporting tool*				
~	none					
✓	medium	manual	online	within a browser	within an authoring tool	install on hard drive
×	fully					
Accessibility approach	Bobby		✓			
	Lift				✓	✓
	A-Prompt					✓
	TAW		✓	✓		✓
	HERA		✓			
	Dante			✓		✓
	Dante within WSDM	×				
	PAN					~
	BW	×				
	WAVE		✓	✓		
	FAE				✓	
	Crunch					✓
	MAGENTA					✓
	Cynthia Says		✓			
	aDesigner					✓
	W3C MVS		✓			✓

Figure 5. Analysis of the population at the supporting tool dimension (2007, Adriana Martín. Used with permission.)



W3C Markup Validation Service

W3C markup validation service was created and maintained by Gerald Oskoboiny under the auspices of the quality assurance activity[26]. Most Web documents are written using markup languages, such as HTML or XHTML. *W3C Markup validator* is an online free service that helps check the validity of these Web documents for conformance to W3C recommendations[27] and other standards. These markup languages are defined by technical specifications, which usually include a machine-readable formal grammar (and vocabulary). The act of checking a document against these constraints is called validation, and this is what the *W3C markup validator* does. The focus of the tool is in validating Web documents as an important step which can dramatically help improving and ensuring their quality. Given an URL or a file upload, the tool validates the document and produces a text report. *W3C markup validator* does not perform a fully quality check but includes usability enhancements, improved feedback, and better support for both W3C and nonW3C document types. The *W3C markup validator* source code is available and a step-by-step guide is provided for the installation of the tool on a server.

Due to the reasons explained below we place *W3C markup validator* at *conformance* from the assessment criteria dimension, *online* but also *install on hard drive* from the supporting tool dimension and *evaluation* from the assessment deliverables dimension. Finally, we say that *W3C markup validator's* test scope is *simple and limited*, *W3C markup validator's* human intervention is *medium* and *W3C markup validator's* report style is *text-based*. Note that in spite of *Bobby*, *TAW, WAVE,* and *Cynthia Says* share the same classification space with *W3C markup validator*, there is a difference among them: while the formers test and monitor for compliance with some accessibility guidelines or standards (WCAG 1.0, 1999; Section 508, 2003), *W3C markup validator* is a tool that evaluates conformance to the technical specification of a Web document written using a markup language.

DISCUSSION

One of the main advantages of having an evaluation framework for classifying accessibility approaches is the feedback that results from comparing them. While working with WAAM we found that this framework is simple to use and allows evaluating any accessibility approach. Even though most of the approaches apply conformance reviews, our framework can also be used to classify those approaches that have developed their own evaluation method. In this sense, approaches that generate an accessible Web page version can be classified as *others* and according with the kind of method they use, that is, an ontology, an heuristic, a markup framework, and so forth. This is the case of *Dante* (Yesilada et al., 2004), since this approach has developed a Web Authoring for Accessibility (WAfA) tool that it is an ontology to enhance the mobility of visually impaired Web travelers. As another case from the supporting tool dimension, our framework considers approaches have an specific tool support but also manual approaches. For example, *Dante* is available to function *within a browser* but also *installed on hard drive* while *Dante within WSDM* (Plessers et al., 2005) is a *manual* approach. This distinction between *Dante* and *Dante within WSDM* can be addressed at the supporting tool dimension. Now, let us to consider a case from the assessment deliverables dimension. In spite of mainly approaches provide *evaluation* and *evaluation & repair* assessment deliverables, our framework also considers approaches that apply *filter & transformation* processes performed by transcoders. By using the former example, *Dante* is classified as *filter & transformation* since the approach transforms a Web page with respect to an annotation process driven by the WAfA ontology.

Tables 2, 3, and 4 show respectively each framework *dimension* (and their associated *factor*) for the fifteen classified approaches. As we can see from Table 2, the *assessment criteria* dimension shows a major concentration of the population for the *conformance* category. This is because accessibility is normally tested based on guidelines like the WCAG (1999) through a conformance testing method (called also standards review). Figure 3 summarizes this situation in a percentage circular graphic.

In Table 3, the *assessment deliverables* dimension shows that most of the approaches correspond to the categories *evaluation* and *evaluation & repair*. This is also related with the fact that accessibility is normally tested based on guidelines and mainly with a tool. Figure 4 summarizes this situation in a percentage circular graphic.

Finally, in Table 4 the *supporting tool* dimension distributes population evenly among the categories. The reason for including this view is because many of the approaches have developed more than one supporting tool. We can see also some prevalence over the *install on hard drive* category followed by the *online* category. Figure 5 summarizes this situation in a percentage circular graphic.

CONCLUSION

In October 1997, the W3C WAI launched the establishment of the International Program Office (IPO)[28]. In the conference press about the launch, Tim Berners-Lee, W3C director, commented (Thatcher, Burks, Heilmann, Henry, Kirpatrick, Lauke et al., 2006):

The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect. The IPO will ensure the Web can be accessed through different combinations of senses and physical capabilities just as other W3C activities ensure its operation across different hardware and software platforms, media, cultures, and countries.

Undoubtedly, Web accessibility is one of the main actors upon which the success of a Web site depends. As a natural reaction to this reality, a diversity of approaches from accessibility research communities and organizations has literally bloomed up.

With this in mind, this chapter introduced the Web accessibility Assessment Model (WAAM), a framework for classifying Web accessibility assessment approaches. After applying the framework to classify 15 different approaches, we found out it a useful model for understanding and discussing Web accessibility as it allows the identification and classification of many different concerns involved when analyzing an accessibility approach.

From the assessment criteria dimension, WAAM not only considers approaches applying traditional conformance reviews but also approaches using a specific developed evaluation method. Meanwhile, from the supporting tool dimension, WAAM addresses approaches with tool support but manual approaches too. Finally, from the assessment deliverables dimension, WAAM provides a space for classifying evaluation and evaluation & repair approaches, but in addition for filter & transformation approaches as well.

REFERENCES

Benavídez, C., Fuertes, J., Gutiérrez, E., & Martínez, L. (2006). 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, ISBN 3-540-36020-4.

Binghamton University. (2001). State University of New York. *Evaluation, repair, and transformation* *tools for Web content usability and accessibility*. Retrieved October 17, 2007, from http://usable.binghamton.edu/currenttools.html#Repair

Bohman, P., & Anderson, S. (2005). A conceptual framework for accessibility tools to benefit users with cognitive disabilities. In *Proceedings of the International Cross-Disciplinary Workshop on World Wide Web accessibility (W4A)*, (pp. 85-89). ACM 2005.

Brajnik, G. (2004). Comparing accessibility evaluation tools: A method for tool effectiveness. *Universal Access in the Information Society, 3*(3-4), 252-263. Retrieved October, 17, 2007, from http://www.informatik.uni-trier.de/~ley/db/journals/uais/uais3.html

Brajnik, G. (2006). Web accessibility testing: When the method is the culprit. In K. Miessenberg et al. (Eds.), *10th International Conference on Computers Helping People with Special Needs (ICCHP 2006),* (pp. 156-163). Springer-Verlag.

Brajnik, G., Cancila, D., Nicoli, D., & Pignatelli, M. (2005). Do text transcoders improve usability for disabled users? In S. Harper et al. (Eds.), *International Cross-Disciplinary Workshop on World Wide Web accessibility (W4A 2006)*, (pp. 9-17). ACM 2006.

Cabinet Office. (2003). Web guidelines for UK government Web sites, from http://www.cabinetoffice.gov.uk/e-government/resources/handbook/html/htmlindex.asp

Cechich, A., & Piattini, M. (2002). Issues for assessing component-based systems. *In Proceedings of the VIII Congreso Argentino en Ciencias de la Computación (CACIC 2002)*, (pp. 253-262).

CLF. (2001). Common look and feel for the Internet—accessibility. Retrieved October 17, 2007, from http://www.tbs-sct.gc.ca/clf-nsi/index_e.asp

Colajanni, M., Grieco, R., Malandrino, D., Mazzoni, F., & Scarano, V. (2005). A scalable framework for support of advance edge services. In *Proceedings of the International Conference on High Performance Computing and Communication (HPCC 2005)* (pp. 1033-1042).

European Union. (2002). *Council of the European Union: Accessibility of public Web sites—accessibility for people with disabilities: Council resolution.* Retrieved October 17, 2007,from http://www.legi-internet.ro/index.php?id=149&L=2

Goble, C., Harper, S., & Stevens, R. (2000). The travails of visually impaired Web travelers. *In Proceedings of the Eleventh ACM on Hypertext and Hypermedia*, (pp. 1-10).

Gupta, S., & Kaiser, G. (2005). Extracting content from accessibility Web pages. *International Cross-Disciplinary Workshop on World Wide Web accessibility (W4A)*, (pp 26-30). ACM 2005.

Harper, S., Goble, C., & Stevens, R. (2003). Traversing the Web: Mobility heuristics for visually impaired surfers. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03)*, (pp. 200-208).

HKSAR. (2001). *Hong Kong Special Administrative Region: Digital 21 strategy. Tips for improving accessibility of Web pages*. Retrieved October 17, 2007, from http://www.info.gov.hk/digital21/eng/knowledge/access_main.html

Hori, M., Ono, K., Abe, M., & Koyanagi ,T. (2004). Generating transformational annotation for Web document adaptation: Tool support and empirical evaluation. *International Journal of Web Semantics, 2*(1), 1-18.

HREOC. (2003*). Human rights and equal opportunity commission.* The Guide to Minimum Web site Standards Accessibility. Retrieved October 17, 2007, from http://www.agimo.gov.au/practice/mws/accessibility/

Iaccarino, G., Malandrino, D., & Scarano, V. (2006). Personalizable edge services for Web accessibility. *International Cross-Disciplinary Workshop on World Wide Web accessibility (W4A)*, (pp. 23-32). ACM.

Irwin, M., & Gerke, J. (2004). Web-based information and prospective students with disabilities: A study of liberal arts colleges. *EDUCAUSE Quarterly*, *27*(4), 51-59.

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.

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

ISO/TS 16071. (2002). International Organization for Standardization/Technical Specification. Ergonomics of human-system interaction—Guidance on accessibility for human-computer interfaces.

Ivory, M., Mankoff, J., & Le, A. (2003). Using automated tools to improve Web site usage by users with diverse abilities. *IT & Society*, *1*(3), 195-236.

Leporini, B., & Paternò, F. (2004). Increasing usability when interacting through screen readers. *International Journal Universal Access in Information Society (UAIS), 3*(1), 57-70.

Leporini, B., Paternò, F., & Scorcia, A. (2006). An environment for defining and handling guidelines for the Web. In K. Miessenberg et al. (Eds.), *10th International Conference on Computers Helping People with Special Needs (ICCHP 2006),* (pp. 176-183). Springer-Verlag.

Loiacono, E. (2004). Cyberaccess: Web accessibility and corporate America. *Communications of the ACM, 47*(12), 82-87.

Martín-Albo, J., Bertoa, M. F., Calero, C., Vallecillo, A., Cechich, A., & Piattini, M. (2003). CQM: A software component metric classification model. *In Proceedings of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2003)*, (pp. 54-60).

Matera M., Rizzo F., & Toffetti Carughi, G. (2006). Web usability: Principles and evaluation methods. In E. Mendes & N. Mosley (Ed.), *Web engineering* (pp. 143-179). Springer-Verlag.

Paciello, M. (Ed.). (2000). *Web accessibility for people with disabilities*. R & D Developer Series, Group West Press.

PAS 78. (2006). *Publicly available specification: A guide to good practice in commissioning accessible Web sites, ICS 35.240.30*. Disability Rights Commission (DRC), 2006. Retrieved October 17, 2007, from http://www.drc.org.uk/library/website_accessibility_guidance/pas_78.aspx

Petrie, H., & Weber, G. (2006). People with disabilities: Automatic and manual evaluation of Web sites (Introduction to Special Thematic Session). In K. Miessenberg et al. (Eds.), *10th International Conference on Computers Helping People with Special Needs (ICCHP 2006),* (pp. 152-155). Springer-Verlag.

Piattini, M., Calero, C., & Genero, M. (2002). *Information and database quality*. Kluwer Academic Publishers.

Plessers, P., Casteleyn, S., Yesilada, Y., De Troyer, O., Stevens, R., Harper, S., & Goble, C. (2005). Accessibility: A Web engineering approach. In *Proceedings of the 14th International World Wide Web Conference (WWW2005)* (pp. 353-362).

Rafla, T., Robillard, P., & Desmarais, M. (2006). Investigating the impact of usability on software architecture through scenarios: A case study on Web systems. *Journal of Systems and Software (JSS2006), 79*(3), 415-426.

Rangin, H. B. (2006). University of Illinois tools and techniques for functional Web accessibility. In K. Miessenberg et al. (Eds.), *10th International Conference on Computers Helping People with*

Special Needs (ICCHP 2006), (pp. 230-233). Springer-Verlag.

Ruiz, J., Calero, C., & Piattini, M. (2003). A three dimensional Web quality model. *In Proceedings of the International Conference on Web Engineering*. (LNCS 2722, pp. 384-385).

Section 508. (2003). *U.S. Federal Government: A quick reference guide to Section 508 resource documents*. Retrieved October 17, 2007, from http://www.accessibilityforum.org/paper_tool.html

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*. Retrieved October 17, 2007, from http://www.pubbliaccesso.it/biblioteca/documentazione/guidelines_study/index.htm

Takagi, H., Asakawa, Ch., Fukuda, K., & Maeda, J. (2004). Accessibility designer: Visualizing usability for the blind. In *Proceedings of the International ACM Conference on Assistive Technologies (ASSETS2004)*, (pp. 177-184). ACM.

Thatcher, J., Burks, M., Heilmann, C., Henry, S, Kirpatrick, A., Lauke, P., et al. (2006). *Web accessibility—Web standards and regulatory compliance.* FriendSoft Press.

WCAG 1.0. (1999). *Web content accessibility Guidelines 1.0. World Wide Web Consortium (W3C) Recommendations*. Retrieved October 17, 2007, from http://www.w3.org/TR/WAI-WEB-CONTENT/

WebAIM. (2006). Web accessibility in mind articles. *Accessibility evaluation tools, 1999-2006*. Retrieved October 17, 2007, from http://www.webaim.org/articles/tools/

Xiaoping, Z. (2004). Evaluation and enhancement of Web content accessibility for persons with disabilities. Doctoral Dissertation, University of Pittsburgh.

Yesilada, Y., Stevens, R., & Goble, C. (2003). A foundation for tool-based mobility support for visually impaired Web users. In *Proceedings of the 12th International World Wide Web Conference (WWW2003)*, (pp. 422-430).

Yesilada, Y., Harper, S., Goble, G., & Stevens, R. (2004). Screen readers cannot see: Ontology-based semantic annotation for visually impaired Web travelers. In *Proceedings of the International Conference on Web Engineering (ICWE2004),* (pp. 445-458).

KEY TERMS

Accessible Barrier: Is any condition that makes it difficult for people to achieve a goal when they are using a Web site through specified assistive technology.

Annotation: "A remark attached to a particular portion of a document, and covers a broad range in literature… Web annotation is crucial for providing not only human-readable remarks, but also machine-understandable descriptions, and has a number of applications such as discovery, qualification, and adaptation of Web documents" (Hori, Ono, Abe, & Koyanagi, 2004, p. 2).

Evaluation and Reporting Language (EARL): A general-purpose language developed by the W3C for expressing test results.

False Positive: An issue detected by a test that after a manual evaluation it is not consider an accessibility barrier.

Issue: "An instance of a potential problem detected by a test" (Brajnik, 2004, p. 257).

Potential Problem: A possible accessibility barrier detected by a test that requires manual evaluation to identify if it is an accessibility barrier or not.

Screen Readers: Special applications that vocalize the onscreen data. Pages are typically read from the top left to the bottom right, line by line, one word at a time.

Transcoder: A Web-server system that produces, on fly, a transformed version of a Web page requested by a user or a browser (Brajnik, 2005).

Travel Objects: The environmental features or elements that travelers use or may need to use to make a successful journey (Yesilada, 2003).

True Problem: An accessibility barrier detected by a test.

ENDNOTES

1 See http://www.w[3].org/WAI/
2 See http://www.sidar.org/index.php or http://www.fundacion.sidar.org/index_en.php
3 See http://www.cast.org/.
4 See http://aware.hwg.org/
5 See http://www.webaim.org/
6 See http://www.utoronto.ca/atrc/
7 See http://www.fundacionctic.org/web/contenidos/es
8 See http://www.watchfire.com/.
9 See http://webxact.watchfire.com/.
10 See http://www.usablenet.com/.
11 At http://www.tawdis.net/taw3/cms/es
12 At http://www.sidar.org/index.php#probhera
13 See http://www.w3.org/Style/CSS/
14 See http://www.php.net/
15 See http://cohse.semanticweb.org/
16 See http://www.w3.org/DOM/
17 A complete list of barriers can be found at http://www.dimi.uniud.it/giorgio/publications.html#hcihw05
18 See http://disabilities.temple.edu/
19 See http://www.uiuc.edu/
20 See http://home.worldonline.es/jlgranad/xhtml/xhtml1.htm
21 See http://www.icdri.org/
22 See http://www.cynthiasays.com/
23 See http://services.alphaworks.ibm.com/
24 See http://www.webstore.jsa.or.jp/webstore/Top/indexEn.jsp
25 See http://www-306.ibm.com/able/guidelines/web/accessweb.html
26 See http://www.w3.org/QA/
27 See http://www.w3.org/TR/#Recommendations
28 See http://www.w3.org/WAI/IPO/Activity