An Accessibility Assessment Framework for Improving Designers Experience in Web Applications

Theofanis Oikonomou[1], Nikolaos Kaklanis[1,2],
Konstantinos Votis[1], and Dimitrios Tzovaras[1]

[1] Informatics and Telematics Institute, 6th Km Charilaou-Thermi Road,
Thermi-Thessaloniki, PO BOX 60361, GR-57001, Greece
Pattern Recognition Laboratory, Computer Engineering and Informatics,
University of Patras, Rio, Patras, Greece
[2] Department of Computing, University of Surrey, Guildford, United Kingdom
{thoikon,nkak,kvotis,tzovaras}@iti.gr

Abstract. The current situation for the accessibility assessment of web applications is encouraging, though not sufficient. Many efforts have been made for the development of various tools that carry out the evaluation of web pages. However, their effectiveness is somehow incomplete. In this work we propose a new framework that has adopted the latest web standard and supports a plethora of assessment configurations to meet the user needs.

1 Introduction

Accessibility is a vital issue nowadays, due to the increasing number of people with disabilities. Despite the fact that Information and Communications Technologies are evolving rapidly, the developed products lack in accessibility. As a result, older people and people with disabilities cannot benefit from ICT based products. The lack of accessible software applications result in economical losses, due to the fact that people with disabilities cannot access the developed products.

Well-defined standards exist that can aid the development of accessible products; unfortunately it is obvious, that developers are not adequately aware of the deficiencies and the boundaries that people with disabilities face while using a software application. Thus, the need for tools that can perform the accessibility evaluation and can aid developers during the development process is vital.

A number of accessibility assessment tools exist, which are responsible for the evaluation of Web sites concerning the accessibility. Some of the most known tools are the WAVE [1], HERA [2], eAccessibility Checker [3], ATRC Accessibility Checker [4] and TAW [5]. Most of these tools perform the accessibility evaluation of Web pages taking into account the Web Content Accessibility Guidelines (WCAG) 1.0 [6], which was developed by the Web Accessibility Initiative (WAI) [7]. But, since December 2008, the old standard WCAG1.0 was replaced by the newest Web Content Accessibility Guidelines (WCAG) 2.0 [8] standard. Thus, the aforementioned tools are not up-to-date with the current accessibility standard. Currently, two

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2011, LNCS 6765, pp. 258–266, 2011.

accessibility assessment tools have partly adopted WCAG2.0, namely ATRC Accessibility Checker and TAW (beta version).

The Web Accessibility Assessment Tool (WaaT) is an integrated tool that provides to users (developers, designers, testers) the opportunity to evaluate the accessibility status of a web application (Web site) according to the WCAG 2.0 standard. Through the tool, designers and developers can be assisted on accessible software application development, through relevant guidelines on accessibility constraints, errors and warnings that should be taken into account. Specifically, WaaT can detect accessibility violations and it can also provide informative tips that will aid the users in the correction of the detected errors/warnings.

The adopted evaluation framework allows users (developers, designers, testers, etc.) to perform a personalized accessibility assessment process, though the selection of different accessibility constraints (e.g. different types of impairments and disabilities, different sets of guidelines, personas). Thus, the evaluation engine of the tool obtains the automatic tests corresponding to the set of accessibility requirements and constraints selected by the user.

The remaining of this paper is organized as follows: We briefly describe the components that the Web accessibility assessment framework consists of in Section 2, while in Section 3 we present the proposed tool followed by some screenshots. Final remarks are discussed in Section 4 which concludes this paper.

2 Web Accessibility Assessment Tool

Six kinds of components are identified in order to support the overall accessibility assessment process of preferable Web applications. Thus, as depicted in the architectural scheme of the following Figure (Fig. 1), the Web applications assessment module consists of the following integrated sub-components:

- **A Web Crawler:** A relevant Web Crawler has integrated into the Web assessment module for creating a copy of the preferable Web applications (Web pages) for later processing by the accessibility assessment core module and also for further validation of the HTML and CSS code. The incorporated crawler "reads" the visible text, the hyperlinks, and the content of the various tags used in the Web application. Using the information gathered from the crawler, as well as from the integrated HTML parser then the overall assessment process can successfully starts. The maximum number of the web pages that should be evaluated by the user can be modified through the Graphical User Interface (GUI) of the tool.
- **The W3C[9] Markup Validator:** Most Web documents are written using markup languages, such as HTML or XHTML. The Web applications assessment tool integrates the W3C Markup Validator in order to return all the potential errors and warnings that can be found in the examined web pages. The W3C Markup Validator is an open-source script written in Perl and for its proper execution it has installed in the ACCESSIBLE web server.

- **The HTML Parser:** An HTML Parser can use Groovy scripts in order to parse the web page source code and get the necessary information concerning the desired elements/attributes of the HTML/XHTML.
- **The W3C CSS Validator:** The W3C CSS Validator Service is a free software created by the W3C organisation to help Web designers and developers to evaluate Cascading Style Sheets (CSS). The W3C CSS Validator has integrated in the Web applications assessment module. The Web Accessibility Evaluator sends to the W3C CSS Validator the URL of the CSS file to be examined and the W3C CSS Validator returns the errors and the warnings that have been found in the CSS.
- **The CSS Parser:** A CSS parser has integrated to the Web applications assessment module which is responsible for the parsing of all the CSS files that are connected with the examined web application.
- **The Web Accessibility Evaluator:** The Web Accessibility Evaluator can be considered as the core component of the Web applications assessment module. It takes as input the URL/local path/source code of the software application that the user wants to evaluate and then it performs the accessibility assessment according to the preferable accessibility rules and constraints. Thus the evaluation engine of the tool obtains the automatic and semi automatic WCAG 2.0 implemented tests corresponding to the set of accessibility requirements and constraints selected by the user.

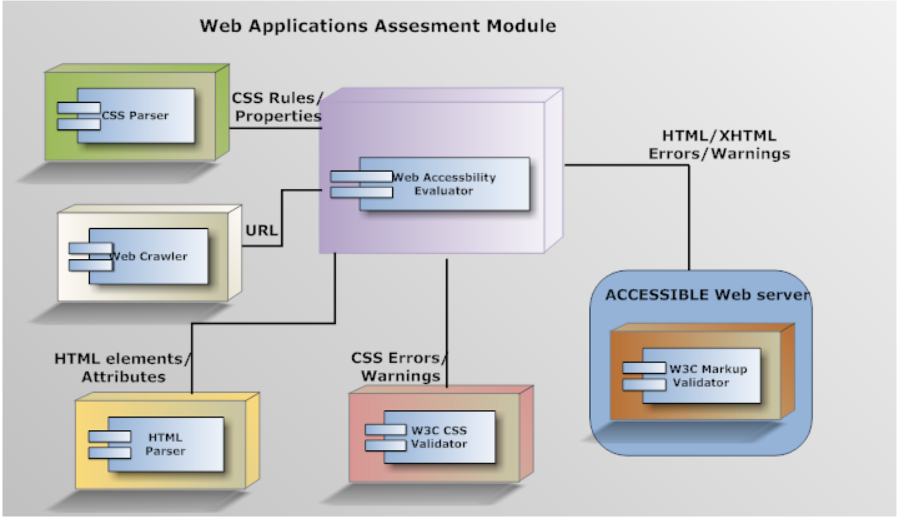
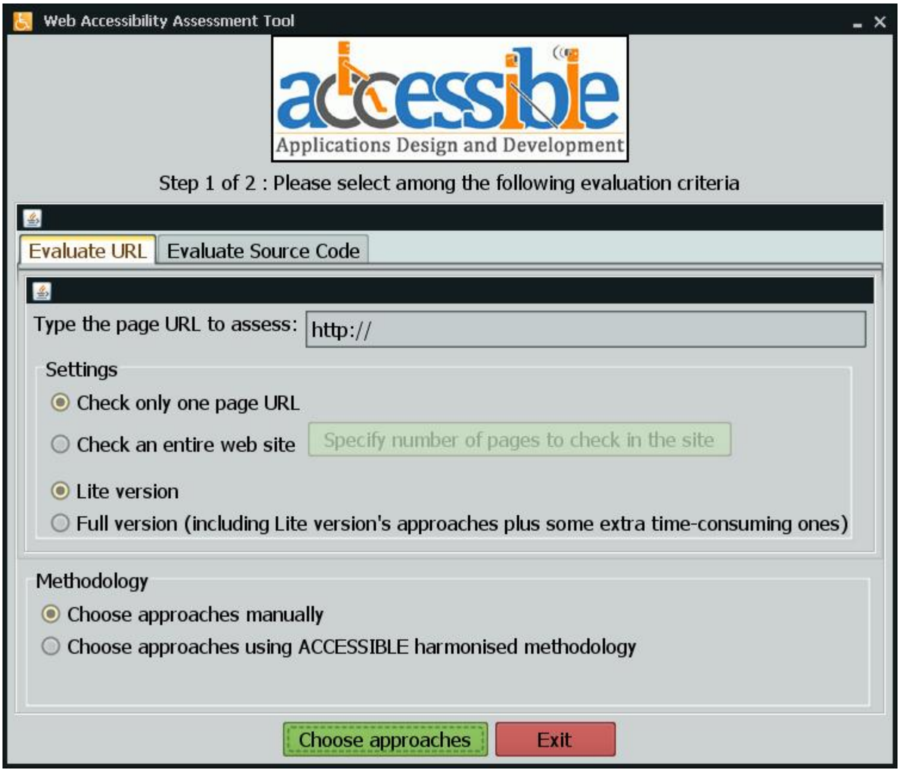


Fig. 1. Block Diagram of the Accessible Web Applications Assessment Module

3 Evaluate a Preferable Web Application

Through the tool, users (designers, developers, evaluators) can be assisted on the design and development of accessible software applications. The adopted evaluation methodology of the tool allows users to perform a personalized accessibility assessment process, through the selection of different accessibility constraints (e.g. different types of impairments and disabilities, different sets of WCAG 2.0 techniques and tests, personas). The GUI of the web accessibility assessment tool is shown in the following image (Fig. 2).

The initial step for the execution of the assessment procedure is the selection of the preferable web page that the user wants to evaluate. It can be specified, whether a specific web page or a whole web site is going to be assessed. At the latter case the number of web pages that are going to be included in the assessment can be determined. Moreover, the tool supports the evaluation of HTML source code.


Web Accessibility Assessment Tool
accessible
Applications Design and Development
Evaluate URL
Evaluate Source Code
Choose approaches
Exit

Step 1 of 2 : Please select among the following evaluation criteria

Type the page URL to assess: http://

Settings

● Check only one page URL

○ Check an entire web site Specify number of pages to check in the site

● Lite version

○ Full version (including Lite version's approaches plus some extra time-consuming ones)

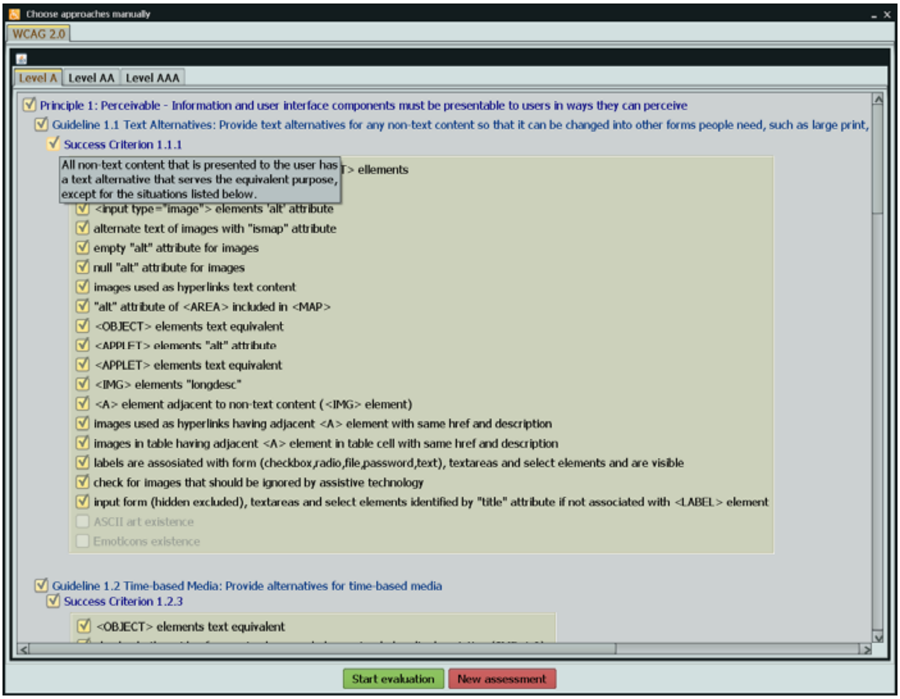
Methodology

● Choose approaches manually

○ Choose approaches using ACCESSIBLE harmonised methodology

Fig. 2. WaaT GUI

After the selection of the preferable web application (web page, web site or HTML source code), the next step involves the selection of the appropriate evaluation methodology that the user wants to use for the assessment. The first method supports the manual selection of the supported approaches for each of the WCAG2.0 success criterion. Thus, for each Success Criterion a number of approaches are suggested and



☑ Principle 1: Perceivable - Information and user interface components must be presentable to users in ways they can perceive
☑ Guideline 1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print,
☑ Success Criterion 1.1.1

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.

T> ellements
☑ <input type="image"> elements 'alt' attribute
☑ alternate text of images with "ismap" attribute
☑ empty "alt" attribute for images
☑ null "alt" attribute for images
☑ images used as hyperlinks text content
☑ "alt" attribute of <AREA> included in <MAP>
☑ <OBJECT> elements text equivalent
☑ <APPLET> elements "alt" attribute
☑ <APPLET> elements text equivalent
☑ <IMG> elements "longdesc"
☑ <A> element adjacent to non-text content (<IMG> element)
☑ images used as hyperlinks having adjacent <A> element with same href and description
☑ images in table having adjacent <A> element in table cell with same href and description
☑ labels are assosiated with form (checkbox,radio,file,password,text), textareas and select elements and are visible
☑ check for images that should be ignored by assistive technology
☑ input form (hidden excluded), textareas and select elements identified by "title" attribute if not associated with <LABEL> element
□ ASCII art existence
□ Emoticons existence

☑ Guideline 1.2 Time-based Media: Provide alternatives for time-based media
☑ Success Criterion 1.2.3
☑ <OBJECT> elements text equivalent

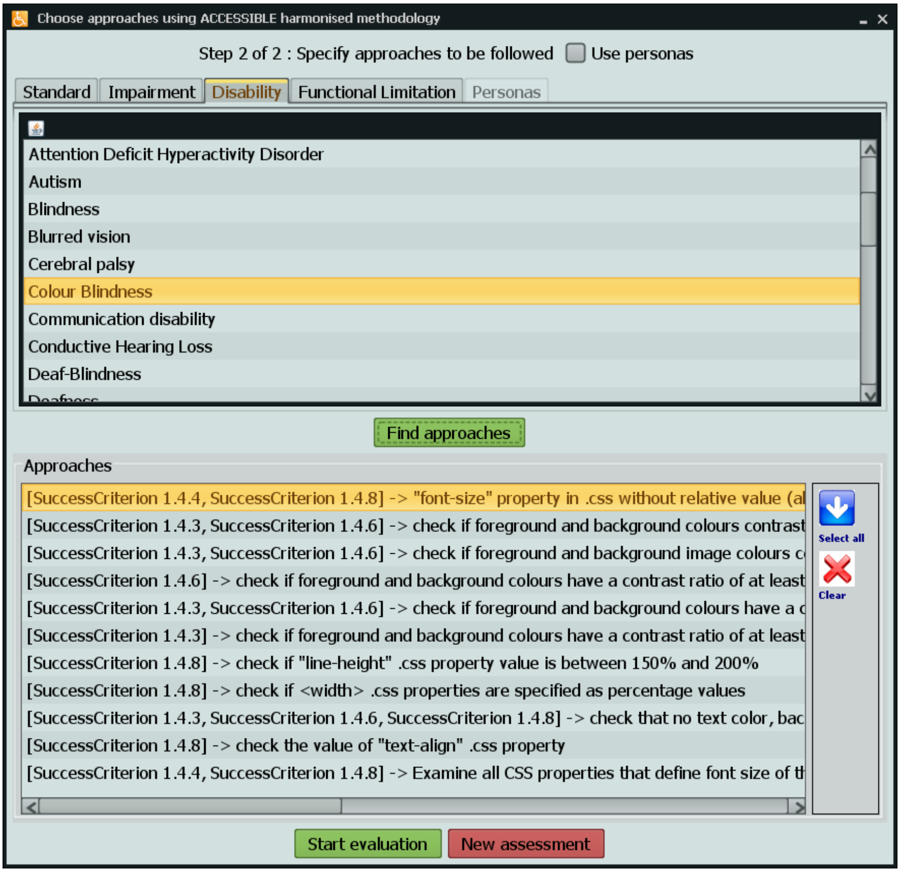
Fig. 3. Manual selection of approaches

they are grouped according to the Level of Priority of each Success Criterion (Fig. 3). By these means, specific or all of the guidelines/success criteria defined in the WCAG2.0 standard can be chosen.

As an additional evaluation method the harmonized methodology [10] defined in the premises of the ACCESSIBLE FP7 project [11] is being supported by the tool. The user can select a preferable impairment and/or disability (Fig. 4). Then the tool maps the chosen impairment/disability to the guidelines of the WCAG2.0 standard and moves on to the evaluation procedure. This way, a personalized accessibility assessment is achieved.

The user has also another possibility in addition to the selection of preferable impairment and/or disability, namely the selection of appropriate Personas (Fig. 5). These Personas are developed to reflect the impairments, as well as combination of impairments, which are met more often. Thus, having already determined the impairments that each Persona suffers from, the tool can perform an assessment that focuses on the specific Persona. WaaT supports the AEGIS-based Personas [12] defined in the premises of the AEGIS FP7 project [**Error! Reference source not found.**] and the ACCESSIBLE-based Personas [14] defined in the premises of the ACCESSIBLE FP7 project.

When the assessment is completed, the evaluation results are presented to the user. These results are grouped according to the priority level of each one of the corresponding Success Criterion (Fig. 6). The results provide detailed information concerning the number of identified errors (red color) and possible warnings (orange color - needs further manual investigation by the user). Furthermore, the assessment results can be saved in PDF or EARL [15] format. PDF is a human readable format and EARL is a machine readable format, that serves to the exchange of evaluation results between Web accessibility evaluation tools. Additionally, detailed information concerning the identified problematic elements is presented, such as the description of the detected error and a suggested tip in order to solve the problem.



Choose approaches using ACCESSIBLE harmonised methodology

Step 2 of 2 : Specify approaches to be followed □ Use personas

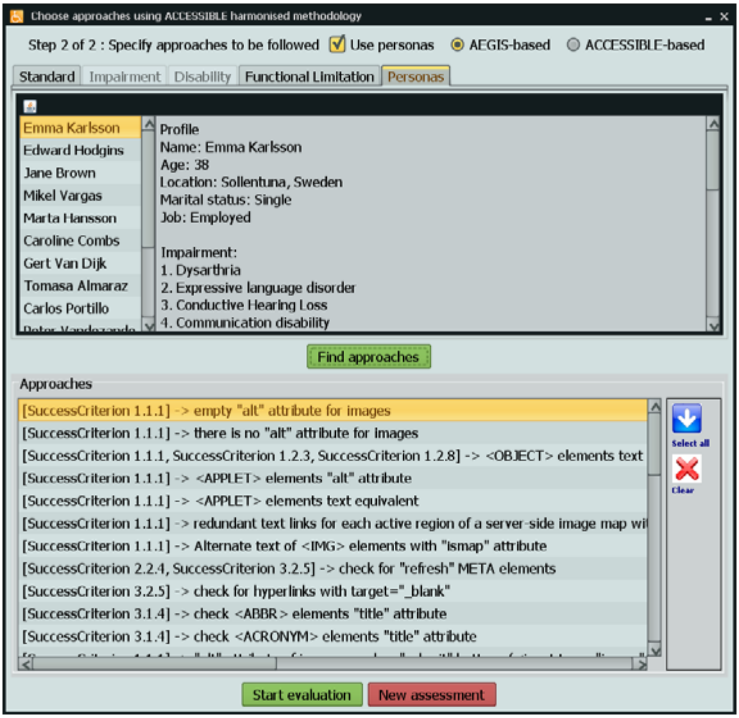
Attention Deficit Hyperactivity Disorder
Autism
Blindness
Blurred vision
Cerebral palsy
Colour Blindness
Communication disability
Conductive Hearing Loss
Deaf-Blindness
Deafness

Approaches

[SuccessCriterion 1.4.4, SuccessCriterion 1.4.8] -> "font-size" property in .css without relative value (al
[SuccessCriterion 1.4.3, SuccessCriterion 1.4.6] -> check if foreground and background colours contrast
[SuccessCriterion 1.4.3, SuccessCriterion 1.4.6] -> check if foreground and background image colours c
[SuccessCriterion 1.4.6] -> check if foreground and background colours have a contrast ratio of at least
[SuccessCriterion 1.4.3, SuccessCriterion 1.4.6] -> check if foreground and background colours have a c
[SuccessCriterion 1.4.3] -> check if foreground and background colours have a contrast ratio of at least
[SuccessCriterion 1.4.8] -> check if "line-height" .css property value is between 150% and 200%
[SuccessCriterion 1.4.8] -> check if <width> .css properties are specified as percentage values
[SuccessCriterion 1.4.3, SuccessCriterion 1.4.6, SuccessCriterion 1.4.8] -> check that no text color, bac
[SuccessCriterion 1.4.8] -> check the value of "text-align" .css property
[SuccessCriterion 1.4.4, SuccessCriterion 1.4.8] -> Examine all CSS properties that define font size of tl

Fig. 4. Corresponding approaches based in the choice of impairment and disability

Lastly, the user can decide, after manually checking the problematic elements, whether the detected warnings are indeed errors or whether they should be ignored by the tool (Fig. 7).



Step 2 of 2 : Specify approaches to be followed ☑ Use personas ● AEGIS-based ○ ACCESSIBLE-based

Emma Karlsson
Edward Hodgins
Jane Brown
Mikel Vargas
Marta Hansson
Caroline Combs
Gert Van Dijk
Tomasa Almaraz
Carlos Portillo

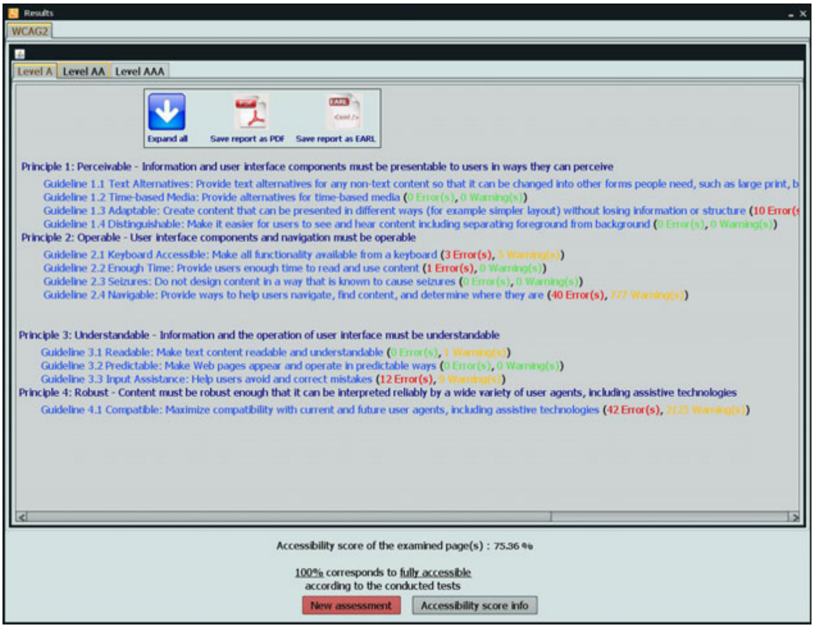
Profile
Name: Emma Karlsson
Age: 38
Location: Sollentuna, Sweden
Marital status: Single
Job: Employed

Impairment:
1. Dysarthria
2. Expressive language disorder
3. Conductive Hearing Loss
4. Communication disability

Approaches

[SuccessCriterion 1.1.1] -> empty "alt" attribute for images
[SuccessCriterion 1.1.1] -> there is no "alt" attribute for images
[SuccessCriterion 1.1.1, SuccessCriterion 1.2.3, SuccessCriterion 1.2.8] -> <OBJECT> elements text
[SuccessCriterion 1.1.1] -> <APPLET> elements "alt" attribute
[SuccessCriterion 1.1.1] -> <APPLET> elements text equivalent
[SuccessCriterion 1.1.1] -> redundant text links for each active region of a server-side image map wi
[SuccessCriterion 1.1.1] -> Alternate text of <IMG> elements with "ismap" attribute
[SuccessCriterion 2.2.4, SuccessCriterion 3.2.5] -> check for "refresh" META elements
[SuccessCriterion 3.2.5] -> check for hyperlinks with target="_blank"
[SuccessCriterion 3.1.4] -> check <ABBR> elements "title" attribute
[SuccessCriterion 3.1.4] -> check <ACRONYM> elements "title" attribute

Fig. 5. Corresponding approaches based in the choice of a specific Persona



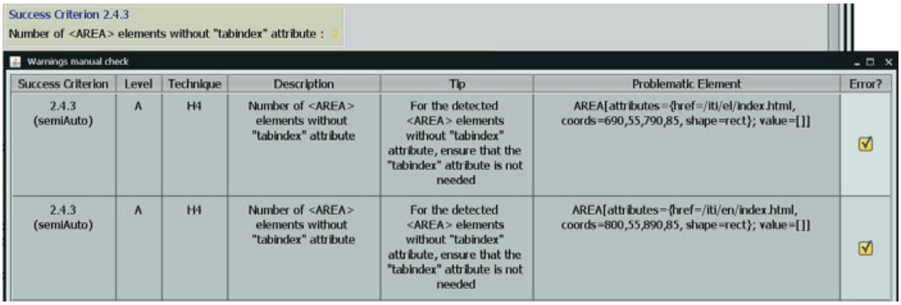
Principle 1: Perceivable - Information and user interface components must be presentable to users in ways they can perceive
Guideline 1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, b
Guideline 1.2 Time-based Media: Provide alternatives for time-based media (0 Error(s), 0 Warning(s))
Guideline 1.3 Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure (10 Error(s
Guideline 1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background (0 Error(s), 0 Warning(s))
Principle 2: Operable - User interface components and navigation must be operable
Guideline 2.1 Keyboard Accessible: Make all functionality available from a keyboard (3 Error(s), 5 Warning(s))
Guideline 2.2 Enough Time: Provide users enough time to read and use content (1 Error(s), 0 Warning(s))
Guideline 2.3 Seizures: Do not design content in a way that is known to cause seizures (0 Error(s), 0 Warning(s))
Guideline 2.4 Navigable: Provide ways to help users navigate, find content, and determine where they are (40 Error(s), 777 Warning(s))

Principle 3: Understandable - Information and the operation of user interface must be understandable
Guideline 3.1 Readable: Make text content readable and understandable (0 Error(s), 1 Warning(s))
Guideline 3.2 Predictable: Make Web pages appear and operate in predictable ways (0 Error(s), 0 Warning(s))
Guideline 3.3 Input Assistance: Help users avoid and correct mistakes (12 Error(s), 9 Warning(s))
Principle 4: Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies
Guideline 4.1 Compatible: Maximize compatibility with current and future user agents, including assistive technologies (42 Error(s), 2125 Warning(s))

Accessibility score of the examined page(s) : 75.36 %

100% corresponds to fully accessible
according to the conducted tests

Fig. 6. Assessment results



Success Criterion 2.4.3

Number of <AREA> elements without "tabindex" attribute : 2

Success Criterion	Level	Technique	Description	Tip	Problematic Element	Error?
2.4.3 (semiAuto)	A	H4	Number of <AREA> elements without "tabindex" attribute	For the detected <AREA> elements without "tabindex" attribute, ensure that the "tabindex" attribute is not needed	AREA[attributes={href=/iti/el/index.html, coords=690,55,790,85, shape=rect}; value=[]]	☑
2.4.3 (semiAuto)	A	H4	Number of <AREA> elements without "tabindex" attribute	For the detected <AREA> elements without "tabindex" attribute, ensure that the "tabindex" attribute is not needed	AREA[attributes={href=/iti/en/index.html, coords=800,55,890,85, shape=rect}; value=[]]	☑

Fig. 7. Information about some detected warnings

4 Conclusions

The novelty that WaaT presents is the adoption of the latest accessibility standard, proposed by W3C, namely WCAG2.0. WaaT can perform a variety of tests based on WCAG2.0 and exceeds the performance of the other tools that have adopted WCAG2.0 during the evaluation process. Moreover, WaaT has implemented a number of tests, in a very precise way, so that no deviations exist between the implemented tests and the proposed techniques and success criteria of WCAG2.0. Thus, the users are assured that the developed applications are inline with the proposed accessibility standards.

Moreover, WaaT proposes a new method of evaluating the accessibility of Web applications, by introducing the harmonized methodology. WaaT is user oriented and can perform personalized evaluation of Web sites, taking into account specific disabilities. Consequently, a Web site can be tested against specific tests that correspond to a disability, resulting in a Web site that is accessible by a specific target group. As a result, people with disabilities are not excluded from the opportunity of accessing the World Wide Web.

Finally, WaaT is a strong accessibility assessment tool that is also easy-to-use. WaaT aids the developers, by providing clear guidance through the development process. Simultaneously, WaaT improves the designer's experience of Web applications and also raises the awareness of people towards the specific needs of people with disabilities.

Acknowledgments. This work was partially funded by the EC FP7 project ACCESSIBLE- Accessibility Assessment Simulation Environment for New Applications Design and Development, Grant Agreement No. 224145.

References

1. WAVE ,`http://wave.webaim.org/`
2. HERA, `http://www.sidar.org/hera/index.php.en`
3. eAccessibility Checker, `http://accessibility.egovmon.no/en/pagecheck/`

4. ATRC Accessibility Checker, http://achecker.ca/checker/index.php
5. TAW, http://www.tawdis.net/
6. Web Content Accessibility Guidelines (WCAG) 1.0, http://www.w3.org/TR/WCAG10/
7. Web Accessibility Initiative (WAI), http://www.w3.org/WAI/
8. Web Content Accessibility Guidelines (WCAG) 2.0, http://www.w3.org/TR/WCAG20/
9. The World Wide Web Consortium (W3C), http://www.w3.org/
10. D3.1 ACCESSIBLE Harmonized Methodology, http://www.accessible-project.eu/documents/ACCESSIBLE_D3.1.pdf
11. ACCESSIBLE FP7 project, http://www.accessible-project.eu/
12. AEGIS Personas, http://www.aegis-project.eu/index.php?option=com_content&view=article &id=63&Itemid=53
13. AEGIS FP7 project, http://www.aegis-project.eu/
14. ACCESSIBLE Personas, http://www.iti.gr/accessible/ACCESSIBLE-ITI-WP4-PU-Del4.1.pdf
15. EARL, http://www.w3.org/TR/EARL10/