myWebAccess: A platform for repairing, enhancing and re-distributing Web Services accessible to people with disability

Georgios Bouloukakis, Ioannis Basdekis, Constantine Stephanidis

MYWEBACCESS: A PLATFORM FOR REPAIRING, ENHANCING AND RE-DISTRIBUTING WEB SERVICES ACCESSIBLE TO PEOPLE WITH DISABILITY

Georgios Bouloukakis[1*], Ioannis Basdekis[2], Constantine Stephanidis[2,3]
[1]*INRIA Paris - Rocquencourt, France*
[2]*Institute of Computer Science Foundation of Research and Technology – Hellas (FORTH)*
Heraklion, GR-70013, Greece
[3]*Department of Computer Science, University of Crete*

ABSTRACT

Web services are an emerging technology which has attracted much attention from both the research and the industry sectors in recent years. The exploitation of web services as components in web applications facilitates development and supports applications interoperability, regardless of the programming language and platform used. However, existing web services development standards do not take into account the fact that the provided content and the interactive functionality should be accessible to, and easily operable by, people with disabilities. This paper presents a platform named *myWebAccess*, which provides a mechanism for the semi-automated “repair” of web services’ interaction characteristics in order to support the automatic generation of interface elements that conform to the de facto standard of the Web Content Accessibility Guidelines 2.0. *myWebAccess* enhances interaction quality for specific target user groups, including people with visual and motor disabilities, and supports the use of web services on diverse platforms (e.g., mobile phones equipped with a browser). The users of *myWebAccess* can create a personalized environment containing their favourite services, and can interact with them through interfaces appropriate to their specific individual characteristics.

KEYWORDS

Web Services, Web Accessibility, User Interface, Standards.

1. INTRODUCTION

The World Wide Web Consortium (W3C) has established the “Web Accessibility Initiative” (WAI-W3C), whose main objective is to provide web accessibility solutions for people with visual, hearing, physical, cognitive and neurological disabilities. The results[1] of this initiative include mainly technical guidelines, such as the Web Content Accessibility Guidelines (WCAG), the Authoring Tool Accessibility Guidelines (ATAG), and the User Agent Accessibility Guidelines (UAAG). In addition, Mobile Web Best Practices provide generic instructions for building applications for mobile devices (Chuter & Yesilada, 2009). WCAG is the most renowned de facto standard, as it provides guidelines on how to create accessible interface and content elements in such a way that they can be read and manipulated by assistive technology solutions. Although these guidelines are mentioned in policies all over the world (e.g., the upcoming European Accessibility Act), they require manual forethought, since existing development tools do not directly support compliant code generation. Except for people with disabilities, WCAG serves those with low level of experience with computers. Finally, it facilitates interoperability with new and emerging technology solutions (e.g., navigator with voice recognition for car drivers).

Despite the proven usefulness of WCAG for web accessibility, it is common for web content manufacturers to ignore or overlook them, thus limiting the ability of disabled users to navigate through the information and services offered by a website. Thus, the aforementioned principles are far from being while integrated, even to public websites where legislation enforces it. Diachronically, studies reveal that web

[1] http://www.w3.org/standards/techs/accessibility#w3c_all

* The Work reported in this paper was conducted when the first author was affiliated with ICS-FORTH, Greece.

accessibility metrics are worsening worldwide (Basdekis, et al., 2010). Web services provide a standard form of communication between different software applications, which in turn support user's interaction through a GUI layout. However, the standards developed so far do not take into account the fact that the content and functionality that a web service offers should be accessible to people with disabilities.

For instance, the typical web service presented in Figure 1 is incorporated as a component to the www.in.gr website and provides the meteorological weather forecast for Athens. By inspecting the relevant "img" tag of the source code, it can be noted that there is no alternative description. Therefore, although it is easy for the "able-bodied" to understand the provided information, this information is not available to a blind user interacting through a screen reader.

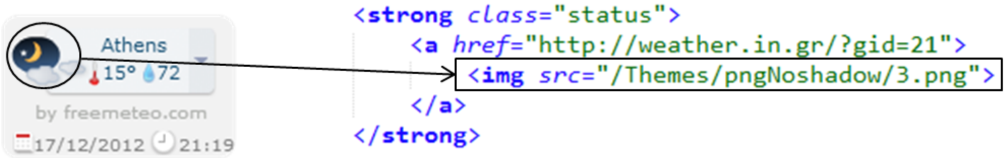

Figure 1. Image without equivalent alternative text to inform the user about the cloudy evening

Aiming to fill in such accessibility gap, this paper contributes practically applicable solutions to ensure that web services have all the presentation characteristics required to render them accessible to people with disabilities. In the context of this work, by analyzing the standards for web services, the description of data was examined, in order to implement rules that indicate the "additional metadata". The final service should be enriched with such "additional metadata" in order to comply with accessibility guidelines when incorporating it into a web site.

The findings from the above analysis provided the specifications for the design and implementation of a system that is able to semi-automatically repair problematic web services, and offer them through a web application in a uniform and user-friendly manner compatible with assistive technology. In addition, this system automatically adapts content generated by third parties and provides it to various browser-equipped devices, using different personalization options for each end user.

Outline. The paper is organized as follows: Section 2 provides an overview of the *myWebAccess* platform and the mechanism for adding and enriching web services with accessibility characteristics. Section 3 discusses the possibility to utilize web services as components to a website, providing adaptation in different context of use in order to support accessible and multi-channel web interfaces. Section 4 presents the evaluation results and section 5 compares *myWebAccess* to related work. Finally, conclusions and future directions are discussed in Section 6.

2. *MYWEBACCESS* PLATFORM

Web services support application interoperability regardless of platform and programming language (Papazoglou, 2008). In the future, some of the most interesting web services will support solutions for applications in order to overcome interoperability challenges of heterogeneous information systems. For example, healthcare applications face the problem of adaptation and interaction with existing systems in a constantly changing healthcare environment. The definition of web services for those applications (Barbarito, et al., 2012), (Mykkanen, et al., 2005) will offer seamless and high quality care to the patients through a web application, especially for people with disabilities.

There are several ways to describe the data transferred via a web service, such as the DTD, XML Schema, RSS and WSDL. WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described at an abstract level, and subsequently are bounded to a concrete network protocol where each message has a specific structure for an endpoint (Papazoglou, 2008). In other words, the WSDL standard corresponds to a web service that specifies exactly the input/output and the necessary procedure to invoke it. The work reported in this paper utilizes the WSDL standard in order to improve content accessibility.

However, even in the case of using the WSDL specification, there are technical limitations. More specifically, developers in most cases fail to provide all the necessary information, since there are cases where WSDL does not cover exhaustively all situations, and as a result presented metadata are limited in order to conform to the specific provisions of the WCAG. An example is a web service which creates an image displaying text given as input by the user. By using the WSDL specification, the input of the web services is represented by some parameters such as height (Integer), width (Integer), color (String) and text (String). In order to fill the above fields the WSDL standard does not completely determine the unit of measurement for the height of the image (pixel or inches). Also, there is no restriction on the font size of the image text. Finally, the color model (e.g., RGB, HSL, CMYK) that should be used is not clear. In order to support the provision of "correctly constructed" web services in practice, and make their usage easy and convenient by people with disabilities, the following should be provided:

- Possibility of adding web services (add).
- Possibility of enriching parameters dealing with presentation or interaction elements (repair).
- Generation of the input and the outcome of the web service in a way that conforms to WCAG (adapt).

Thus, it seems necessary to create an information system able to scan the available functionality of a web service and include the related components. This system should provide a mechanism to add metadata in order to augment web services with accessibility features. Additionally, the preferences stored in the user profile will further facilitate the interaction of end-users with the services and will provide easier access per user category (e.g., profiles for people with dyschromatopsia). At the same time, and in accordance with recent technological advances, any web service should be able to adapt to user device's capabilities.

In the context of this work, a platform called *myWebAccess* is proposed and implemented. The platform provides a mechanism for adding and semi-automatically "repairing" third-party web services, so that all the interaction elements can be determined and thus content generation complaint to WCAG 2.0 can be achieved.

2.1 Towards ensuring accessibility characteristics in Web Services

Web services can be classified in two categories according to their interaction with the user. The first category includes web services that present information (**output**) without any user input, while the second requires implicitly or explicitly some **input** prior to the presentation of the outcome. For example, a weather forecast is a service of the second category that requires the name of the desired city through text input. Taking into account the interaction behavior of a user utilizing assistive technology, the input/output of such service should provide several technical features that enhance accessibility. More specifically, and besides adhering to WCAG 2.0, a convenient navigation mechanism needs to be provided (e.g., extra features for motor impaired users). Moreover, balanced color contrast between foreground and background should be used to increase text readability, with potential fluctuation of the font size. In the case of non-text context, an equivalent alternative text description should be provided for images, graphics and multimedia. Data tables must involve caption and summary tags. In the case of interactive forms, each "input" tag should have labels to be properly announced and navigational aids should speed up navigation between fields. In addition, a mechanism for validating user input should be provided for error prevention. Finally, additional adaptation features can be triggered depending on the device used (e.g., width of a mobile screen).

Summarizing, the input/output of a web service should be suitably adapted to the specific preferences of the user and the technical capabilities of any screen in use. To improve the integration of a web service in an application, appropriate metadata characterizing the input/output elements should be present. This additional information (**metadata**) should be provided somehow (e.g., by the manufacturer or at a later stage by a service administrator) in order to overcome limitations. In order to create a web service which will contain all the necessary interaction information to appropriately interact with the user, the implementation of the following steps was deemed necessary: (i) analysis of the WSDL file, (ii) separation of the parameters that needed the additional metadata, (iii) import of metadata for each parameter based on the WCAG and (iv) creation of an XML file that defines the additional metadata.

As depicted in Figure 2, the generated XML is directly linked to the originated WSDL, and both files will provide the necessary information for the exploitation of accessible web services. To evaluate the effect of the above procedure, the ideal scenario would be to analyze a web service and request suggestions from users experienced in web accessibility issues. These suggestions would indicate the additional metadata needed. Thus, an experienced web service administrator is needed to repair web services that contain poor or insufficient interaction metadata.

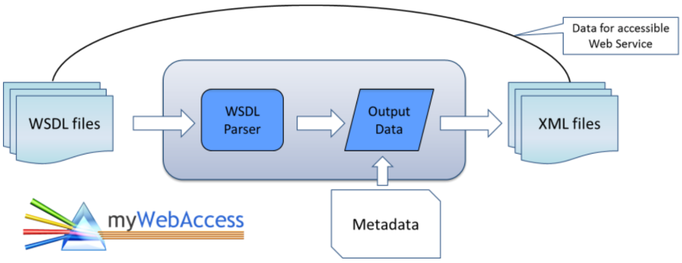

Figure 2. Procedure to add accessibility features

According to the steps described in Figure 2, the process followed for semi-automated repair of a web service and its functions is the following. The WSDL file is given as an input to a parser (step ① in Figure 3) that defines a variety of meta-data for each data type (i.e., input/output parameter) and correspond to a web service. Next, the additional metadata are analyzed[2] (step ② in Figure 3) and presented to the administrator by using a web interface that indicates the "gaps" that should be filled for each input/output parameter (step ③ in Figure 3). Next, the system generated or manually inserted metadata are stored in an XML file (step ④ in Figure 3). The WSDL and the additional metadata (XML file) are directly connected to each other, so that presentation elements conforming to the WCAG can be generated (step ⑤ in Figure 3). Finally, the end users are able to interact with services, since they are integrated with valid and compliant mark-up (step ⑥ in Figure 3). This process ensures the proper enrichment of the parameters prior to the publication and availability of the Web service and its functions.

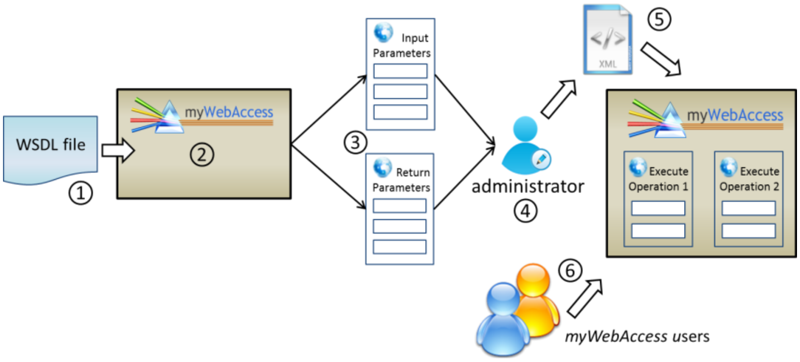

Figure 3. Process to import a web service to the *myWebAccess* platform

The exploitation of accessible web services as components in a web application has the following advantages: (i) **reuse**: reduce manufacturing costs and availability (Clark, 2003), (ii) **multi-device**: easier adaptation of the presented information to various devices, (iii) **conformity with standards**: achievement of interoperability with assistive technology solutions and, (iv) c**ost**: lower upgrade costs accordance to the technological development (Sierkowski, 2002).

3. ACCESSIBLE AND MULTI-CHANNEL WEB INTERFACE GENERATION

The majority of web applications adopt a specific content structure (Curtis, 2009). More specifically, according to common practice followed in recent years, there are four main areas (i.e., Header, Sidebar, Main content and Footer as shows the design template ① in Figure 4). Those areas usually have the same look & feel on all pages of a website. Furthermore, this structure can easily reutilize different design templates. By setting specific content areas in advance, the initial structure (design template ① in Figure 4) can be enriched with additional navigation aids (design template ② in Figure 4), or in the case of a display on a mobile device, the initial structure can split into more pages for improved readability (design template ③ in Figure

[2] http://www.urdalen.no/wsdl2php/index.php

4). In both cases, the developer has to ensure the proper adaptation of the main content that is updated frequently. As mentioned earlier, web services can be treated in a web application as components where functionality is provided by third parties (Figure 5), resulting in easier adaptation to different design templates. Thus, appropriate mechanisms are necessary to present Web services for different third parties in a unified presentation schema.

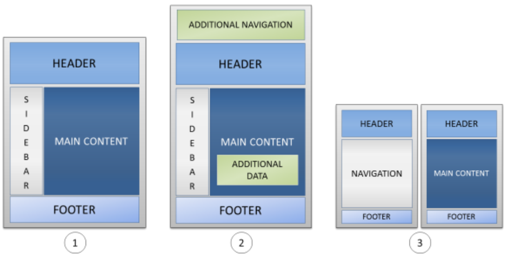

Figure 4. Various design templates for websites

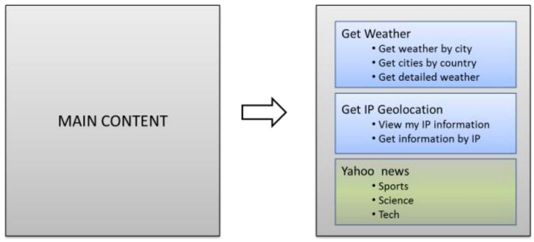

Figure 5. Web services as components to a website

After the insertion of a web service by the system administrator, the *myWebAccess* platform provides automatically generated content (step ⑤ in Figure 3) that is compliant to the WCAG 2.0 technical standards. This content is enriched with semantic information (if required), is adaptable to support specific interactions, and has the ability to invoke this service. Thus, taking advantage of the above, *myWebAccess* creates a suitable interface for each user category by serving: (i) **blind users** with content that conforms to the WCAG 2.0, level AA, (ii) **users with impaired vision** or **color blindness** with high contrast background - foreground, (iii) **motor impaired users** with sliding navigation (Adams, et al., 2007), (Myers, et al., 2002) and a virtual keyboard (Norte & Lobo, 2007), (Zhai, et al., 2000) and, (iv) supporting **mobile devices**.

The purpose of most information systems is to retrieve data from a source and display it to the user. If the "user interface" and the "application logic" are connected to a single part/object of the application, then whenever the user requires a change in the user interface, the whole section containing the "logic" is modified too (Fielding, 2000). The Model-View-Controller (MVC) architecture (Reenskaug, 2003) addresses this issue by separating the display of the data, the functionality of the application and the data storage. The *myWebAccess* platform has been built using the MVC architecture to support multiple design templates. More specifically, the View is responsible for the display of data and the Model for storing this data. Finally, the Controller undertakes the handling of a user request, namely, the data recovery and the selection of the appropriate design template.

The architecture of the *myWebAccess* platform provides the necessary degree of freedom to address the diversity of user requirements and devices, covering the different scenarios derived from the combination of these two factors. This is achieved through a library of alternative design templates and the classification of users into different profiles. For instance, in order to facilitate motor impaired users, the library contains a suitable design template and provides sliding navigation and a virtual keyboard (Adams, et al., 2007), (Norte & Lobo, 2007), (Figure 6).

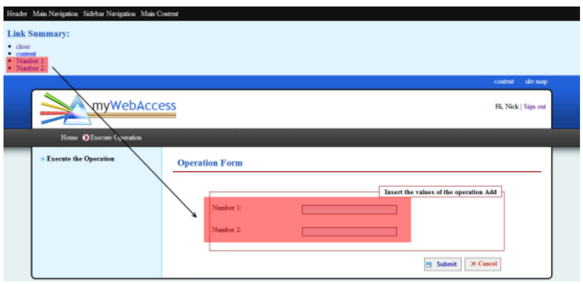

Figure 6. Sliding navigation design template with a virtual keyboard

This enables the user to navigate the web page by using one button. For example, in order to fill in the field named "Number 1", the sliding window contains the corresponding hyperlink and through it the virtual keyboard is displayed. Then, by using only two buttons (Norte & Lobo, 2007) the user can fill in the field. It is worth noting that without the repairing of services by the *myWebAccess* platform (Figure 3), the mark-up of the operation "add" would not be valid, and therefore it could not be invoked by using only two buttons.

4. EVALUATION

To assess the level of compliance of the services available to the end-user with WCAG 2.0, *myWebAccess* was tested using semi-automated accessibility testing tools. Three categories of web services were tested: (i) news feeds via RSS, (ii) meteorological weather forecast, and (iii) image generator. More specifically, proper display was checked on different browsers and devices, the usage of meta-language content was validated using the W3C validation tool and compliance with WCAG 2.0 was evaluated using the Web Accessibility Test (T.A.W.) tool[3]. Regarding the validation of the used meta-language (XHTML), the generated web pages are constructed according to the W3C standards of XHTML and CSS. All the generated content complies with the accessibility guidelines WCAG 2.0 for all web pages of the *myWebAccess* platform. The results show that there were no problems observed, except for 30 notes. These notes cannot be controlled by the tool and require the manual inspection of a specialist. For instance, a specialist has to check the description of the images and the page title that should be descriptive. Furthermore, the headers on each page, the labels and the forms should be well structured and quite descriptive. The combination of the selected colors and the degree of the contrast, as well as the existence of alternative navigation on the website, are all tested and were found to comply.

After making any adjustments arising from the results of the above-mentioned tests, the second step consisted of evaluating the usability for end users. In this context, a scenario of use was elaborated for evaluating proper service performance, as well as the usability and usefulness of the above services for each user category. This scenario contains three tasks for the selection, management and invocation of "repaired" services in the *myWebAccess* platform. Ten experts in the use of assistive technology were involved using profiles of visual or motor impaired users. All users completed the scenario within an acceptable period of time. The maximum time was 22 minutes. As depicted in the diagram of Figure 7, those using a screen reader were slower compared to the rest of the subjects and carried out tasks in two to four timeframes. This difference in performance was expected, since in this case the user is forced to listen to an important part of the content before understanding at which point that part is, while filling in data forms requires a special procedure. However, on the whole, everyone finished their task within reasonable timeframes.

The number of errors made by each user was also measured. Figure 8 illustrates in detail the number of errors that experienced users made throughout the process. As expected, a high number of errors were observed in the category of users who use a screen reader. Especially in the case of the scenario simulating the behavior of a user with upper limb disability by using binary switches, a remarkable difference was noticed in the execution time compared to the rest of the users. The tasks were performed by two users with prior experience in the use of these devices (binary switches). In this use case, an additional navigation with fast access hyperlinks (Sliding Navigation) was activated (Figure 6).

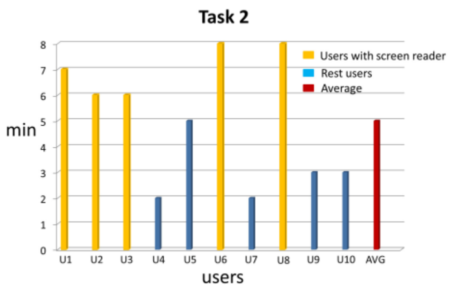

Figure 7. Required time (min) for the completion of task

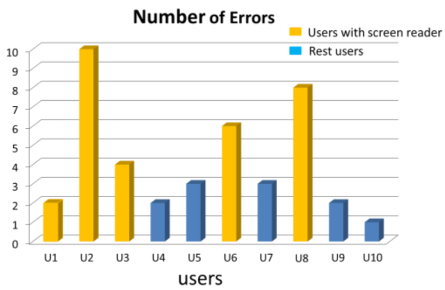

Figure 8. Number of errors for each user

Figure 9 shows that, for each task usage scenario, the results of the execution time between the user 1 and user 2 are similar. Compared to the rest of the users (Figure 9), the time is almost four times greater. This difference occurs due to the fact that the users were asked to fill data in forms using the virtual keyboard (with sequential selection of characters).

[3] http://www.tawdis.net

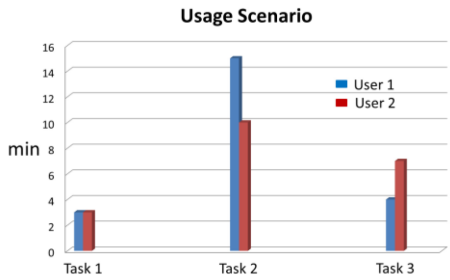

Figure 9. Time (min) for completing three tasks

5. RELATED WORK

This section presents various efforts by the research community to address various related issues.

Accessibility in the Web Services domain. To achieve the exploitation of web services by all users, including people with disabilities, a tool has been proposed, for assessing the accessibility degree of a web service (Giakoumis, et al., 2011). By using this tool, it is possible to assess whether a SOAP or REST web service conforms to the WCAG. Based on the functionality of the above tool and due to the limited requirements of the WSDL standard, coupled with the common practice of developers that does not fully exploit the specifications during development, a SOAP type web service will rarely be assessed as fully accessible. Also, in the case of an input REST type web service, and due to the absence of a standard for this type (REST), the tool provides an interface through which a user is able to create a specification called WADL (Giakoumis, et al., 2011). The problem of this procedure is that the user puts significant effort with the risk not to define the necessary interaction metadata based on WCAG. Finally, the above tool provides only information about the accessibility degree of a web service, without proposing a solution to repair them.

Furthermore, a system architecture based on services has been developed (Kehagias, et al., 2011), which offers easier navigation in a city for motor impaired users. Based on the functionality of this system, it was noticed that the specifications which define the format of data in a WSDL are quite limited, especially in the case of annotating them with semantic information.

Personalized Interfaces. An optimal environment for all internet users would be a web application that will collect all third party services which conform to WCAG 2.0 in order to interact with assistive technology solutions. Additionally, the possibility of customizing the system for certain categories of users, including people with vision or motor impairment, would provide the necessary degree of personalization to achieve a friendly interface. There are many popular web applications that provide personalized interfaces with the favourite gadgets of each user as components (i.e., iGoogle, Netvibes, MyYahoo and Gritwire). None of those applications uses third party services for their functionality, except some commercial services such as iGoogle, without conforming their interface to WCAG.

Automatic adaptation. Besides offering online applications that collect third party services, there are systems that use various methods to analyze a website in order to convert the extracted content to another format. These applications usually operate as a proxy by making an analysis of the web mark-up. Then, they reclassify the content while making some sort of corrections to continuously improve its accessibility. For example, BBC offers a service named Betsie which transforms the content of the website, making it easier to read for blind and color-blind users (Brown & Robinson, 2001). Moreover, the WebFACE tool (Alexandraki, et al., 2004), through which extra features are added to enhance the accessibility. However, it is applied only to specific web page structures (Maeda, et al., 2004). Finally, a system for dynamically updating web pages to achieve a high contrast background-foreground, delete images and parts of the context that do not conform with WCAG 2.0 and offer an alternative navigation is presented in (Richards & Hanson, 2004). All mentioned systems, by using some form of heuristic algorithms, parse the content and then adjust and rearrange it by using appropriate tags. The success of the final result depends on how the initial version of the website conforms to valid XHTML meta-language.

6. CONCLUSIONS AND FUTURE WORK

The main idea behind the work presented in this paper is the exploitation of web services as components in a web interface in order to facilitate the development of accessible and multi-channel web interfaces. A technical process has been elaborated in order to enable accessibility characteristics in the Web Services domain. Leveraging on this process, the *myWebAccess* platform for repairing, enhancing and re-distributing web services accessible to people with disability has been developed. Moreover, the use of the platform has illustrated the adaptation of the repaired services in different context of use. Finally, the proposed approach has been evaluated by examining web services reusability and interoperability with assistive technology solutions.

Following an extensive user-based evaluation, the following extensions could be incorporated into the system in the future: (i) exporting to a script with XHTML valid mark-up by taking the metadata from the existing WSDL and XML files which have been created by the *myWebAccess* platform, (ii) the exploitation of the Semantic Web (Berners-Lee, et al., 2001), and Semantic Web Services (McIlraith, et al., 2001) towards a more comprehensive solution that means transition from the WSDL standard to a standard with semantic information, (iii) including the Accessible Rich Internet Applications Suite (WAI ARIA[4]) to improve the interaction via assistive technology (faster and easier) and, (iv) creating different registries of "repaired" services by leveraging QoS aspects (O'Sullivan, et al., 2002) to support varied disability requirements.

REFERENCES

Adams, C. et al., 2007. *Web standards creativity: innovations in web design with XHTML, CSS, and DOM scripting.*

Alexandraki, C., Paramythis, A., Maou, N. & Stephanidis, C., 2004. Web accessibility through adaptation. *Computers Helping People with Special Needs,* pp. 626-626.

Barbarito, F. et al., 2012. Implementing standards for the interoperability among healthcare providers in the public regionalized Healthcare Information System of the Lombardy Region. *Journal of Biomedical Informatics.*

Basdekis, I., Klironomos, I., Metaxas, I. & Stephanidis, C., 2010. An overview of web accessibility in Greece: a comparative study 2004-2008. *Universal Access in the Information Society,* 9(2), pp. 185-190.

Berners-Lee, T., Hendler, J., Lassila, O. & others, 2001. The semantic web. *Scientific american,* 284(5), pp. 28-37.

Brown, S. & Robinson, P., 2001. *A world wide web mediator for users with low vision*.

Chuter, A. & Yesilada, Y., 2009. Relationship Between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG).

Clark, J., 2003. *Building accessible websites Indianapolis: New Riders*.

Curtis, N., 2009. *Modular Web Design: Creating Reusable Components for User Experience Design.*

Fielding, R., 2000. *Architectural styles and the design of network-based software architectures*.

Giakoumis, D., Tzovaras, D. & Hassapis, G., 2011. Enabling accessibility characteristics in the web services domain. *Universal Access in Human-Computer Interaction. Design for All and eInclusion,* pp. 177-186.

Kehagias, D., Giakoumis, D. & Tzovaras, D. An ontology-based service-oriented application for mobility impaired users.

Maeda, J., Fukuda, K., Takagi, H. & Asakawa, C., 2004. Web accessibility technology at the IBM Tokyo Research Laboratory. *IBM journal of research and development,* 48(5.6), pp. 735-749.

McIlraith, S., Son, T. & Zeng, H., 2001. Semantic web services. *Intelligent Systems, IEEE,* 16(2), pp. 46-53.

Myers, B. A. et al., 2002. *Using handhelds to help people with motor impairments.* New York, USA, ACM, pp. 89-96.

Mykkanen, J. et al., 2005. Designing web services in health information systems: From process to application level. *Studies in health technology and informatics,* Volume 116, p. 515.

Norte, S. & Lobo, F., 2007. *A virtual logo keyboard for people with motor disabilities.* pp. 111-115.

O'Sullivan, J., Edmond, D. & Ter Hofstede, A. What's in a Service?. *Distributed and Parallel Databases*, pp. 117-133.

Papazoglou, M., 2008. *Web services: principles and technology.* s.l.:Addison-Wesley.

Reenskaug, T., 2003. The Model-View-Controller (MVC) Its Past and Present Trygve Reenskaug, University of Oslo.

Richards, J. & Hanson, V., 2004. *Web accessibility: a broader view.* pp. 72-79.

Sierkowski, B., 2002. *Achieving web accessibility.* s.l., pp. 288-291.

Zhai, S., Hunter, M. & Smith, B. A., 2000. *The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design.* New York, NY, USA, ACM, pp. 119-128.

[4] http://www.w3.org/WAI/intro/aria