

Social4all: Definition of specific adaptations in Web applications to improve accessibility



Rubén Gonzalez Crespo ^{a,*}, Jordán Pascual Espada ^b, Daniel Burgos ^a

^a Universidad Internacional de La Rioja (UNIR), Logroño, Spain

^b Department of Computer Science, University of Oviedo, Asturias, Spain

ARTICLE INFO

Article history:

Received 14 December 2015

Received in revised form 14 March 2016

Accepted 3 April 2016

Available online 12 April 2016

Keywords:

Web accessibility

Accessibility evaluation

Automatic evaluation tools

Web adaptation

Web application

ABSTRACT

Web accessibility guidelines help developers to create websites which can more easily be used by people with different limitations. The principles and techniques of accessibility focus on the suitable use of standard Web components, alternative methods to present information, and alternatives to facilitate user interaction. Currently, the biggest part in creating accessible websites is played by Web developers, because they manage the page code. Unfortunately, there are millions of websites which do not follow accessibility guidelines, as this usually requires great effort and knowledge of accessibility issues. This research aims to create a platform, based on a novel approach, which allows a set of accessibility problems to be solved without modifying the original page code. The proposed platform is able to analyse websites and detect many accessibility problems automatically; after this, a guided assistant is used to offer adequate solutions to each detected problem. The assistant tries to abstract references to Web implementation issues and to explain every accessibility problem in an understandable way for non-technical people. This new approach could be useful to improve the level of accessibility of many websites for people besides Web developers.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Web accessibility is a key issue to ensure that people with different disabilities can use the Web. Some people with disabilities have a lot of problems perceiving and interacting with Web applications [1]. Web accessibility principles and techniques focus on the suitable use of standard Web components, the use of alternative methods to present information, and alternative systems to facilitate user interaction. Web accessibility includes recommendations related to text size and style, presentation of multimedia information, description of links, use of mouse events, navigation systems, page organization, colours and contrast, forms, pop-ups, and many other Web elements. (See Table 1.)

These accessibility measures are very useful because some people with disabilities could find the use of some features of websites very complex. Therefore, the main aim is to facilitate the use of websites for persons with low vision, deafness, speech disabilities, cognitive limitations or neurological disabilities. Millions of people in the world have these kinds of problems and must still be able to use the Web [2]. According to European Union data, between 10% and 15% of the population of the European Union 27 have some type of disability [3].

There are some important guidelines in existence which can help developers to create websites with good accessibility. One of the most

extensive sets is WCAG 2.0, which covers a wide range of recommendations for developing accessible Web content. WCAG 2.0 is an evolution of WCAG, which was published in May 1999. W3C recommends compliance with this guide. According to WCAG criteria it is possible that some websites require a greater level of accessibility than others; for this reason WCAG proposes three different accessibility levels from lowest to highest [4] Level A, Level AA and Level AAA.

The Section 508 1194.22 Standards Guide is another popular set of guidelines [5,6], which provides recommendations on many of the issues also covered in WCAG 2.0.

Unfortunately, millions of websites do not follow accessibility guidelines, as this can require great effort and knowledge about accessibility on the part of the Web developer. It is a very complex task to estimate how many websites have accessibility problems. There are millions of sites on the Web and accessibility reports encompass only an insignificant part of them: most accessibility reports focus on popular networks, or networks which provide institutional services, e.g., for hospitals, universities, and governments. In many countries accessibility guidelines must complement government or public websites as a legal obligation. Many reports and academic studies have focused on an analysis of the fulfilment of accessibility guidelines in some specific areas such as education [7,8] or government services [9].

The most part of the responsibility for Web accessibility falls upon Web developers, who need to apply the recommended techniques during implementation and maintenance processes [10,11]. They need to make extra efforts to ensure that websites are suitable for different

* Corresponding author at: Universidad Internacional de La Rioja (UNIR), C/Almansa 101, Madrid 28224, Spain.

E-mail address: ruben.gonzalez@unir.net (R.G. Crespo).

Table 1
Load times for the original website and the same website through the Social4all platform.

Evaluated scenario	Dom ready (ms)	Everything loaded (ms)
<i>Original website</i>		
Without cache	79	208
With cache	12	24
<i>Website in Social4all – no adaptations defined</i>		
Without cache	163	284
With cache	56	83
<i>Website in Social4all – ten adaptations defined</i>		
Without cache	177	299
With cache	79	130

types of users, including users who may be unable to perceive information and navigation systems in the same way as others. Accessibility has to be analysed and tested like other website requirements [12].

The team involved in the development of a website has to know the accessibility guidelines inside out, and to be aware how to present information in the right way, along with alternatives for people with disabilities. Developers also have to know how to properly use HTML, CSS and JavaScript elements. To evaluate the accessibility of a website, developers can carry out a set of validations and checks throughout the development process or during the testing phase.

There are different methods to evaluate accessibility issues; the most effective is the use of accessibility experts to examine all the website and evaluate whether the code and user interfaces meet all the requirements specified by an accessibility guideline, for example WCAG Level AA. Guidelines outline all the issues that developers have to check in websites; many of these need to be interpreted carefully as they can be expressed in an ambiguous way. For example, developers should check whether alternative descriptions of an image describe the image correctly or not.

The team in charge of checking website accessibility can use certain defined tests [13,14]. Some research works have proposed specific tests, surveys or procedures to assist developers in responding to accessibility issues [15,16].

Some tools can detect certain accessibility problems automatically. These tools can also offer some assistance in human evaluation, for example searching a website for potential errors, which must then be evaluated manually later [17–19]. These tools are not enough to automate solutions of accessibility problems but are a great help to save time in the process of finding such problems [20].

Web accessibility faces important challenges, including in particular the following: 1) There are a lot of websites on the Web which do not meet accessibility guidelines; 2) Many elements of accessibility techniques are highly subjective, for example alternative descriptions may be suitable for people with certain limitations, but not for others.

These two problems have no unique and direct solution, because they are motivated by a wide range of different factors. Websites might not follow accessibility guidelines due to various reasons: sometimes the Web developers are not committed to accessibility; in other cases they do not have enough time to check all accessibility issues, or may even not be familiar with the accessibility concept. The creation of automatic tools to assist developers in accessibility evaluations could help in accessibility adoption in many websites.

Many problems related to the subjectivity of some accessibility issues can be solved if the website has more than one version. It could be very useful for example for a website to have a version with alternative descriptions for people with limited vision and another version for people with cognitive limitations; probably, the information needed in order to understand the content will not be the same for people with different limitations [21].

Another measure to improve the accessibility level of websites is the promotion of the implication of more people in the accessibility adoption. Currently, a lot of organizations and old people assistants are

interested in Web accessibility but they cannot choose to make their websites accessible because website developers are the only people with the necessary expertise. Organizations need the collaboration of website developers, and this commitment can be very difficult to obtain in many cases. Also another other profiles related to technology could be interested in use this kind of services like SEO consultants and industries 4.0 experts.

The main objective of this research is to create a novel platform which allows a set of accessibility problems to be solved for any website, without the need to modify the original code of the website, as sometimes it is difficult to include the original developers in tasks related to Web accessibility. The proposal intends to enable the detection of different types of non-accessible content, and to provide an adequate mechanism to fix these accessibility issues. The mechanism to fix the problems has to be suitable for people without programming skills. The platform must try to abstract references to Web technologies, and must explain every problem in an understandable way that is accessible to people without technical expertise. In many cases people who belong to organizations know how to explain the concepts for people with limitations – but perhaps these people do not have technical knowledge. For example: a person may be able to define an alternative description for a picture and provide a good summary of a table of data in a specific way for people with cognitive limitations, but the same person may not have Web development knowledge. Finally, one version of a website may be inadequate for people with different limitations to use the website in a comfortable way. Therefore, a dynamic solution must allow the creation of different accessible versions of the same pages and resources, this multiple-version system offering new opportunities to adapt the Web content for different situations and profiles.

The structure of the paper is as follows. Section 2 presents related works. Section 3 provides a description of the proposed platform. Section 4 presents a use study of the platform. Section 5 presents an evaluation of the results. Finally, Section 6 presents conclusions and future work.

2. Related works

Web accessibility is very important in contemporary society. Companies, organizations and governments have the responsibility to make online information accessible to all kinds of people [21].

Web accessibility has been important for 20 years [22]. Several important sets of accessibility guidelines have been developed since the first guidelines were compiled by Gregg Vanderheiden in 1995 [23]. Accessibility guidelines express requirements which developers' websites have to meet [24]. One way to ensure these requirements are met is by carrying out a human evaluation of the website, but this can often be a complex task, as many websites have a huge amount of content. Some tools can help developers evaluate accessibility automatically or semi-automatically [25].

There is a wide range of professional tools available to evaluate websites' accessibility. These tools are often very popular among Web developers. Some have been around for as long as 12 years [26]. Some of the most popular automatic accessibility evaluation tools are AChecker [19], 508 Checker [27], Evalura [28], Access Monitor [29], Access Lint [30], Cynthia Says [31], SortSite [32], WAVE – Web Accessibility Evaluation Tool [18], Eval Access 2.0 [33] and some others with similar features. These tools apply algorithms to analyse and detect accessibility errors in Web code.

Most of these tools are in a process of continuous improvement, so they commonly include regular new features or mechanisms to detect accessibility problems. After analysis most of the tools show a report with information about detected problems. Some research works focus on normalizing the results of accessibility analysis [34]. Most of these tools classify the results of the analysis in errors and warnings since in many cases it is not possible to know with certainty whether pages contain some types of accessibility errors, because the diagnosis

of these errors often requires human interpretation [20]. However, automatic analysers are very useful to show potential errors or warnings, which developers then have to check manually.

Commercial tools are not a unique alternative. Other tools have been proposed, which often are not widely used, mostly not even having a functional prototype, but which often propose novel ideas. WAEX: is an accessibility evaluator based on XSLT [35]. Like the professional solutions, these tools focus on the reporting of problems, not the solutions.

Only a small number of tools have been proposed which actually suggest possible solutions to detected accessibility problems. These tools require access to the original code and also a person with developer skills. Vu [36] propose a tool which can analyse problems and generate possible solutions, but these solutions also have to be implemented in website code. In their proposal, Mayol and Nadal [37] use a similar approach: their tool analyses the HTML code and in some cases it is able to propose another HTML. The proposed method can be used to replace the original code and solve the problem. None of the proposals which are able to suggest solutions are developed enough to be properly tested; most of them are mainly theoretical approaches.

Some research has demonstrated that adaptations on the client side could lead to good approaches to improve accessibility [38]. However, there are as yet no tools which integrate the client-side modifications to fix the accessibility problems dynamically without the need to modify the original code. The use of this type of adaptations also could allow the creation of multiple versions of the same website. If a novel approach could use a mechanism to fix accessibility problems by abstracting the technical issues it could be used by a wide range of people, for example people belonging to various organizations, occupational therapists, teachers, and friends or relatives of people with limitations.

3. Proposed platform

Social4all is a platform which has the main objective of improving access to online information for people with different kinds of limitations.

The proposed platform allows users to create a specific adaptation profile for any website. The conceptual schema of the platform is shown in Fig. 1. The user can introduce the URL of the website and define a new adaptation profile, which will analyse the website with the use of automatic algorithms in order to track accessibility problems in accordance with a set of guidelines specified in WCAG Level A, Level AA and Level AAA.

After code analysis this platform will show a list of all the accessibility problems found on the page. For every problem the platform will show a description of the accessibility rule which is unfilled, and also a description of the problem explaining why it is important and how the user can fix it. In fixing every accessibility problem the user is required to input some information into the platform. The information requested depends on the type of problem; for example, if the platform detects that the alternative description of a picture is missing then it will ask the user for that alternative description. If the platform detects that the website does not specify the language content, the assistant will show a list of languages for the user to select one from the given options.

Every time that the user defines a solution for one accessibility problem the information related to the solution will be stored automatically in the database. The platform calls each stored problem solution an “adaptation”; every “adaptation profile” will have a set of “adaptations”.

When the user wants to define the “adaptation profile” for a website, the profile will be available and ready to use. Any user can open an adaptation profile. When an adaptation profile is opened, the original code of the website is loaded into the platform, and it will then load all the adaptations defined in the profile and make the appropriate modifications in the copy of the website code; these modifications are made using JavaScript.

Users who have loaded the adaptation profile will view the original website and the adaptations defined by the profile owner. The navigation process is based on the use of a Web gateway, which loads the original code and modifies it.

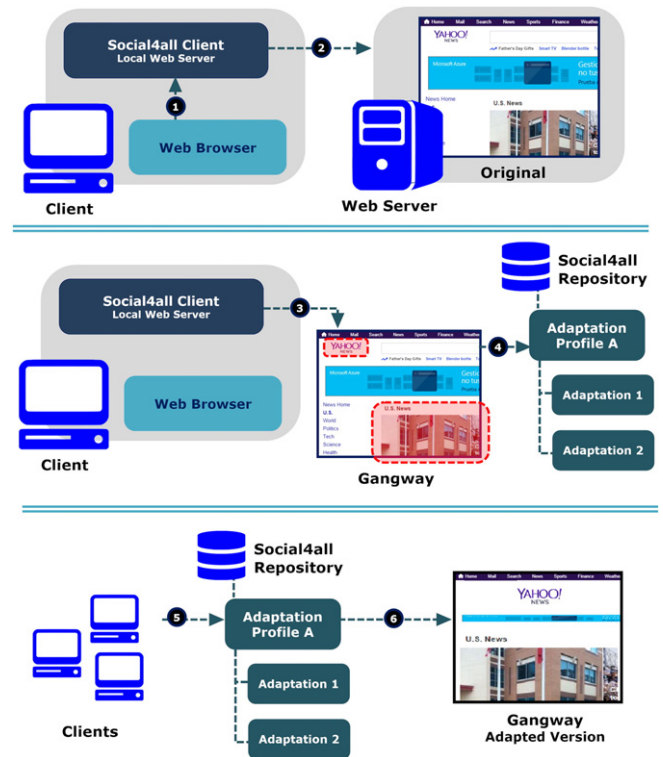


Fig. 1. Social4all conceptual scheme. (1) The user accesses the local Web server and executes the Social4all Client. (2) The Social4all Client gets the original Web code from the URL. (3) It loads the Web code in the Social4all gangway and starts the accessibility analysis. (4) For every detected problem the user can define an adaptation. All defined adaptations in the profile are saved in the Social4all global repository, which is common for all users. (5) Any user can open Social4all Client and use any created profile. (6) The client loads the original code in the Social4all gangway and applies the adaptations which modified the original content of the website.

3.1. Flow and operation

The Social4all platform is composed of a central database and a website client which contains a set of interrelated modules (Fig. 2).

The client of Social4all is a Web application which runs in a Tomcat Server. Every user who wants to use Social4all has to download the client and install it. Using a simple installation assistance in the user machine a local server and an instance of Social4all client will be deployed automatically, so the user can access the Social4all client using any Web browser.

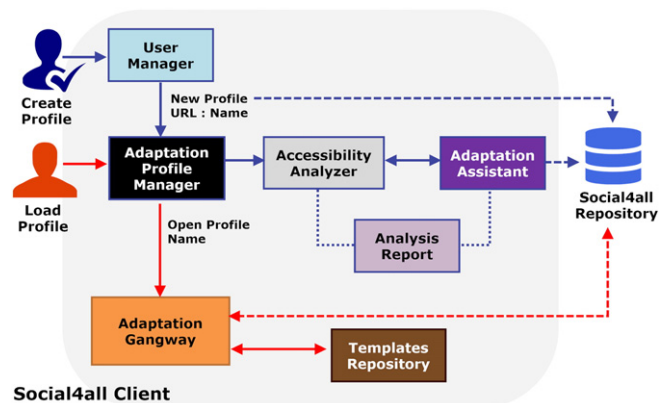


Fig. 2. The architecture of the Social4all client and the relationship between its modules. The diagram also shows the modules involved in the creation of a profile, and in loading profile.

The Social4all client contains the following main modules:

3.1.1. User manager

Users have to be identified in Social4all. User identification is not necessary to apply adaptation profiles, but it is necessary to create new adaptation profiles. Every adaptation profile has a user who is considered as the owner; this user can manage all the issues of the adaptation profile.

3.1.2. Adaptation profile manager

This module allows registered users to create a profile and save all related information. To create a new profile the system needs a name and short description of the profile and the main URL of the website. Using the adaptation profile manager the user can modify at any time all the profiles that he or she has created. The user can define adaptations into an adaptation profile using the module “Accessibility Analyser”. The adaptation profile manager is also useful to view all accessibility profiles stored in the Social4all platform. The platform can apply these profiles using the module “Adaptation Gangway”.

3.1.3. Accessibility analyser

When a user has created a new accessibility profile and wants to add new adaptations, he or she needs to use the “Accessibility Analyser”. As a first step of the process the accessibility analyser does an analysis of the page code. It includes a set of algorithms which can identify accessibility problems. These algorithms are based on guidelines WCAG Level A, Level AA and Level AAA. For example, these algorithms are able to detect multimedia elements without alternative descriptions, placeholder descriptions, pages with no title, input elements without a label, embedded content without alternative transcriptions, links without textual content, tables without captions, improper use of frames, and many other inadequacies. The WCAG guidelines which were considered are: 1.1.1, 1.1. Others, 1.2.8, 1.3.1, 1.3.Others, 2.4.1, 2.4.4, 2.4.6, 2.4.9, 3.1.1, 3.1.2, 3.2.5, 3.3.2, 4.1.1, 4.1.2, and the Techniques for WCAG: H2, H24, H36, H39, H44, H48, H56, H64, H71, H74, H93, F30, G18, G145, G170, G171, G83, C12, C14, C28.

The algorithms analyse the HTML, CSS and JavaScript of the website using jQuery. As a result of the page analysis the analyser generates a report which contains a list of the accessibility problems detected; every problem has its own id. This report will be automatically sent to the module “Adaptation Assistant”. With the use of following code snippet the report shows algorithms detecting two accessibility problems: 1) text inputs without a label, and 2) tables without a summary.

```
$( "input" ).each(function(index) {
    if ( $(this).attr('type') == "text") {
        var idElement = $(this).attr('id');
        if( isDefined(idElement)){
            var label =
                $( "label[for='"+
                    $(this).attr('id')+"']" );
            var labelContent = label.text();
            if (!isDefined(labelContent)) {
                detected_fail +=
                    $(this).attr('social4all') +
                    "input" + 3 + " ";
            }
        } else {
            detected_fail +=
                $(this).attr('social4all') +
                "input" + 3 + " ";
        }
    }
});

$( "table" ).each(function(index) {
    var attr = $(this).attr('summary');
    if (!isDefined(attr)) {
        if( containsSocial4allAttr($(this))) {
            detected_fail +=
                $(this).attr('social4all') +
                "table" + 9 + " ";
        }
    }
});
```

Before the analysis process starts the platform executes a script which gives a unique Id (Social4all Id) to every HTML element on the website. During the analysis it may thus locate the specific element in the Web page, because the accessibility problems will be related to one or more HTML elements and the platform needs to know the exact position of the code which contains the problems.

3.1.4. Adaptation assistant

This module receives the report showing the accessibility problems detected in the page. The assistant then shows the user all detected problems. For every problem it shows its location in the page, a description, and an explanation of how the user can fix the problem. For example, if an alternative description of a picture is required, the form will ask for the description; if the default language of the page needs to be set the form will allow the user to select the appropriate language from a list; and if the page uses content HTML elements for presentation the form will give the user an option to remove the elements which are used improperly.

In every case the user has the chance to define the solution that he considers more appropriate for each problem; mostly, the solutions are not unique because accessibility issues are usually relative to human interpretation. Solutions could depend on the audience of the profile; different audiences could require different types of descriptions, depending on their limitations. The main audience of the profile is also an important factor by which the user might choose to include different type of descriptions. For example: for people with cognitive limitations it could be a good approach to explain carefully all complex concepts, or if the profile is specially designed for a person with low vision it would be a good approach to include alternative descriptions with pictures and videos.

3.1.5. Adaptation gangway

This module is responsible for obtaining the code of the original website and applying it in all the adaptations defined by the user in the profile. From the technical point of view this module has a high complexity level. The adaptation gangway makes a request to the original website and modifies the original code.

As a first step of the process the adaptation gangway changes all the absolute links and paths in the original code. The new paths maintain the same URL as a parameter but the request will be addressed again to the gangway; in this way the petition will continue referencing the original resource but the request will always go through the gangway page; for example: `gangway.jsp?u=www.sample.com/img.png`. Using this system, users cannot leave the gangway page when they open a link, because with this modification all links are redirected to the gangway.

Another important part of the adaptation gangway is the redirection filter. In the original code of the website all the relative resources are referred to the original server, but now the copy in the code is in the client Social4all server, and the server does not have these resources (Fig. 3). The approach taken in solving this kind of petitions will be based on using a redirection filter in the Social4all server. The redirection filter receives all the relative requests from the gangway and composes the equivalent petition for the original server to obtain the requested resource. The module includes another sub-module which aims to construct and define the headers of each http request appropriately. The Web requests are directed to the local Social4all server, but after this the Social4all client has to construct an equivalent request and send it to original website. The information obtained has to be managed because it will include important changes in the http headers, for example maintaining or validating a Web session.

When the website code is finally loaded in the gangway page, the adaptation process begins. The filter will ask for the adaptation list which the user had defined in the adaption profile. Every adaptation contains an Id type, the Social4all Id of the HTML element in the page which contains the accessibility problem, and additional data which will be used to fix the problem. For every adaptation registered in the list, the system gets the adaptation JavaScript and jQuery codes from

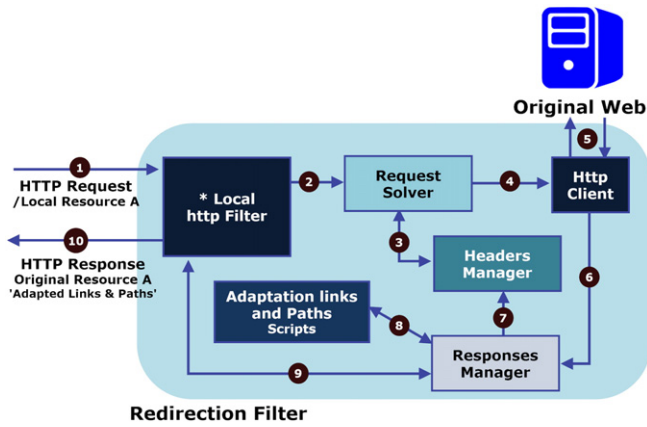


Fig. 3. The diagram shows the process that the filter goes through when it receives a new http request. (1) The request is captured by the local filter. (2) The filter has to find what path is the equivalent resource in the original website; in this task the module has to recalculate the path, considering the base URL and path modifications. (3) It appropriates http headers in the request (content, cookie, session, etc.). (4) It sends the request to the original server. (5) The server generates the response. (6) It receives the server response. (7) It updates the current headers according to the new response. (8) If the content of the response is HTML code it has to apply scripts to modify the links and the paths. (9) It generates the response. (10) The browser receives the response.

the module “Adaptations Repository”. The adaptation repository contains templates of JavaScript and jQuery code to fix a set of accessibility problems. The template does not define all the code to fix the problem; some parts of the code are “parameterized” and will be completed to get functional code using the information defined by the user, for example, the title of the page, a Boolean parameter which defines whether an improper HTML element must be deleted or exchanged, etc.

3.1.6. Templates repository

This module contains the templates of JavaScript and jQuery code to fix a set of accessibility problems. Every accessibility problem included in Social4all is identified by a unique Id and has its own template.

Every template has a different implementation; most focus on the modification of the HTML code, for example adding new attributes to an element, changing the value of an attribute, adding new elements, and exchanging elements for others.

The template includes some parameters which will be replaced for the adaptation information defined by the user. The following snippet of code shows the raised solutions for: 1) text inputs without a label, 2) tables without a summary and 3) inadequate use of content element for presentation.

```
case 3:
%>
var input =
$("*").find("[social4all=<%= s4allid %>]");

var inputId = input.attr('id');

if( isDefined(inputId) ) {

    $("<label>", {
        "for": inputId,
    }).append("<%=adaptationInfo%>")
        .insertBefore("#"+inputId);

} else {

    input.attr('id', input.attr('social4all'));
    var newInputId = input.attr('id');
    $("<label>", {
        "for": newInputId,
    }).append("<%=adaptationInfo%>")
        .insertBefore("#"+newInputId);

}
<%
break;

case 9:
%>
$("*").find("[social4all=<%= s4allid %>]");
attr('summary', "<%=adaptationInfo%>");
<%
break;

case 18:
%>
if( "<%=adaptationInfo%>" == "checked" ){
    $("*").find("[social4all=<%= s4allid %>]")
        .remove();
}
<%
break;
```

The Social4all client has been developed using JavaServer Pages, Java, Servlets, Web Services, JavaScript, JQuery, HTML and CSS. The Web application runs in a Tomcat 7 Server, and is connected to a MySQL Database.

3.2. Accessibility evaluation

The accessibility analyser includes JavaScript algorithms in detecting a set of accessibility problems. Some accessibility problems defined in the WCAG guidelines can be detected automatically, but others need human interpretation to determine whether they are accessibility problems or not.

The adaptation assistant contains a wizard which presents explanations of accessibility problems and additional information on solving them. There are four main wizards, depending on the required information:

1. Short, concrete information about an element. For example the Id of an element.
2. An element description. For example an alternative description of an image.
3. Selecting an option from a set of options. For example page language.
4. Confirming a change in an element. For example asking permission to change an incorrect HTML element.

The adaptations repository includes templates to fix the problem described in the accessibility analyser. We include a set algorithm to detect common accessibility errors which are contained in a lot of websites. The platform architecture is designed for the purpose of enlarging the list of accessibility errors, with the ability both to detect and fix them. The architecture of the platform allows analysis algorithms to be included as well as templates for addressing errors in a very easy and convenient way.

The current list of accessibility problems includes: no alternative description in multimedia elements, pages without a title, inputs without a label in forms, embedded content without alternative content or transcription, pages without default language, links (anchors) without textual content, duplication of unique Ids, tables without a summary, tables with more than one caption, table headers without scope, fieldsets without legends, placeholder content in alternative descriptions, image form buttons without an alternative description, legends without content or placeholder content, map areas without an alternative description, figures without a caption, content elements used wrongly for presentation, links which open new tabs without user permission, incomplete or wrong use of abbreviations, wrong use of iframes, and others.

4. Use case: creating an adaptation profile

In this use case we are going to define two different accessibility profiles for a website. The analysed website contains few accessibility and usability errors and we are going to fix most of them using Social4all.

To create a new adaptation profile the user has to start the Social4all client and open the <http://localhost:8080> address in any Web browser. The user has to log into Social4all and start the creation of a new adaptation profile.

The profile includes an identifying name, the main URL of the website, and the description (Fig. 4). After this, the main page of the website will be loaded in the platform and the user can start the automatic analysis to detect accessibility errors in the page. Users can start the accessibility analysis on any page on the website; another window will open for this.

A checklist will be shown where the user can select whether he or she wants to apply a set of automatic accessibility adaptations. Below the list the window shows a wizard. Each page of the wizard focuses on the explanation and solution of one detected accessibility problem (Figs. 5 and 6).

Fig. 4. The user has to introduce basic information to create a new adaptation profile.

The automatic analysis is continued for the three pages of the website, and after this we define the appropriate solutions for all the errors. In this case the analysed website contains 34 detectable errors.

The user can remove any element of the page by pressing the right button of the mouse (Fig. 7). Removing content could be useful to

simplify the website, or to create profiles focused on specific tasks, for example, a profile for viewing only the latest news in Yahoo!, or a profile for viewing exclusively lists of companies in LinkedIn. Sometimes the simplification of underused content can be an important help for people with limitations.

When the user finishes the definition of the adaptation of the profile it is time to use or share it with other users through a link. The user can modify the profile any time.

Any user of the Social4all client can open any adaptation profile. Using the client every user can write one URL and view all the adaptation profiles related to it. When the user selects an adaptation profile the original website is brought up combined with the adaptations defined in the profile (Fig. 8). The user has the impression he or she is using the real website, but in reality it is the adaptation gangway. Not all the modifications are visible at the front end of the website; some of them are only visible in the HTML code, because some adaptations are useful for screen readers and other applications designed for people with limitations.

5. Evaluation

During the evaluation process we have analysed in detail the most used tools to improve accessibility – AChecker, 508 Checker, Evaluera, Access Monitor, Access Lint, Cynthia Says, SortSite, HERA, WAVE – Web Accessibility Evaluation Tool, Eval Access 2.0 – but none of them is able to fulfil the main objectives of this research. Many of these tools are useful to detect a set of accessibility problems in an automatic

Fig. 5. After doing the automatic analysis in the page index.html the browser opens a wizard with the detected errors. Users can read the error explanation and have the chance to solve it. The screenshot shows the wizard page for error 7: the user can introduce text for the image link; the assistant also shows the element with a potential problem using a red border.

Fig. 6. If the user opens another page on the same website the client starts the automatic analysis process again. The screenshot shows the wizard for the error 2 in the page table.html: the user has the chance to introduce text for the label; the assistant also highlights the label with problems using a red border.



Fig. 7. In the profile definition the user can hide elements of the page to create a simpler version, or avoid irrelevant parts of the website which include a lot of complexity.

way, but not to fix these problems. None of these tools are able to improve the accessibility level without accessing an application server and modifying the source code.

To evaluate the use of the proposed platform by people without Web development skills we carried out a test with 16 occupational therapy students. They were interested in accessibility principles but did not

have much knowledge about Web development. The profiles of these people could be closest to the profiles of real users of the proposed platform.

5.1. Part 1

In the test we asked all participants to create a new adaptation profile for a news website. They had to include at least ten different adaptations in the profile. The students had to try fix all the accessibility errors detected on the website in the best way possible. Then, they had to evaluate four statements about their experience; for this we used a Likert scale (Likert, 1932). The statements given to the students were the following:

- n1: The platform allows easy creation of a new adaptation profile.
- n2: The results of the accessibility analysis are easy to understand.
- n3: The method proposed by the assistant to solve every problem is easy to understand.
- n4: The overall process to create an adaptation is easy to understand.
- n5: The use of adaptation profiles could be an effective approach to improve accessibility.

5.2. Part 2

In the second part of the evaluation we asked the students to create a profile which simplified the content of the page, maintaining only the important content, the news. After this activity they were asked to evaluate four statements about their experience:

- n6: The method proposed to hide elements is easy to use.
- n7: The method proposed to redisplay hidden elements is easy to use.
- n8: Hiding some page elements could be useful to simplify the use of the page in many situations.

The graph Fig.9 shows the average results obtained in the process and the graph Fig.10 shows the standard deviation.

The results show that the satisfaction of the users with the platform was very high. All of them strongly agreed with the statements “The platform allows the easy creation of a new adaptation profile”, “The use of adaptation profiles could be an effective approach to improve accessibility” and “Hiding some page elements could be useful to simplify the use of the page in many situations”. The other statements were also favourably evaluated. The lowest result was for “The method proposed by the assistant to solve every problem is easy to understand”, although this statement obtained an average rating of 4.56/5 – also a positive result in any case the standard deviation was higher than 0.62 which show the students answers were quite uniform.

Average Rating

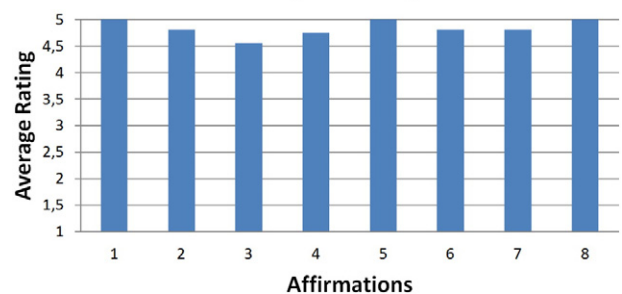


Fig. 9. Average rating of the 16 students' evaluations on the Likert scale. The horizontal axis shows the Id of each statement. The vertical axis shows the rating: 1 – Strongly disagree, 2 – Disagree, 3 – Neutral, 4 – Agree, 5 – Strongly agree.



Fig. 8. In the adaptation profile search form the user can introduce the main URL of a website. The Social4all client shows all the adaptation profiles related to this website. When the user selects the profile the base URL will be open in the gangway, and the user can interact in a normal way with all the pages on the website as if the site was the original, but it contains the modification as defined in the profile.

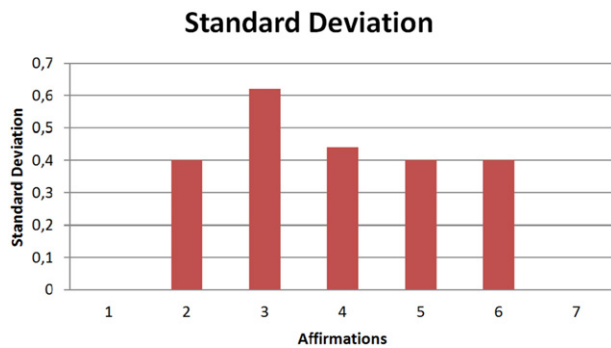


Fig. 10. Standard deviation rating of the 16 students' evaluations on the Likert scale.

In Section 4 we have shown how the proposal can fix some accessibility problems using an adaptation profile. For this process users do not need Web development skills or access to the source code of the website. The process does have a computational cost, however, because the Social4all platform requires some background work: the original code of the website must be requested and loaded in the gangway, and the JavaScript templates need to be applied to include the defined adaptations.

The following table shows the average load times in the original website and the same website opened through the Social4all platform with and without adaptations. The profile contains 14 adaptations, among which include: alternative a descriptions of images, removing HTML items which were used for presentation, include a title in the HTML, include a language tag, include a text alternative to a video, include an abbreviation description, include textual content in a link without any text, include labels in input fields, include captions and summary in a table, disable some links which opens new tabs without a real reason.

For the evaluation process we used the Chrome Web browser on a computer with the specifications Intel (R) Core (TM) i5-2410M CPU @ 2.30GHz, 8.00 GB Ram, OS: Windows 8.1 Pro N × 64.

The obtained results show that the use of the Social4all platform increased the time needed to load the Dom and the full page. However, the platform consumed a reasonable time for the high computational load required: in neither case did it exceed a time of waiting of 300 ms. The heaviest background process was the loading of the page in the gangway: around 75 ms/84 ms without cache and 44 ms/59 ms with cache. Once the page was loaded in the gangway the application of the adaptations was relatively fast, around 14 ms/15 ms without cache and 23 ms/47 ms with cache. The cache is not very useful in the execution of the adaptations because the biggest part of the process is on the client side.

The load time of Social4all could be improved with a few code optimizations; however, these do not relate directly to the research objectives, but rather to Web engineering.

6. Conclusions and future work

This research work aims to create a novel platform which allows a set of accessibility problems to be solved for any website, without the need of modifying the original code.

The proposal allows the detection of part of the non-accessible content of a website; after this, the proposed assistant can be used to offer an adequate solution to each detected problem. This assistant has been devised to be suitable for people without experience of Web development. The assistant tries to abstract the references to Web technologies, and explain every problem in an understandable way for non-technical people. During the evaluation process we tested the proposal with a set of users with no Web development knowledge and the results were satisfactory. The users agreed with the usefulness of the proposal.

The research has therefore achieved its objectives. The novel approach presented in this platform could provide a useful tool to improve the accessibility of many websites, and allow the resolution of accessibility problems for people with different profiles.

There are still issues to be addressed in the proposal, and some commercial accessibility validators are still able to detect more accessibility problems. As next steps, we are going to include new algorithms to detect more accessibility problems and the corresponding mechanisms to solve these problems. Another future research line will seek to include a social network in the platform, to offer the possibility of sharing or commenting on profiles and adaptations.

Acknowledgements

This work is partially funded by the project 2013-I-IDD-00179, called "Platform to adapt accessibility in social networks – Social4All" (ADER 2013), and by the UNIR Research Support Strategy 2013–2015, under the TELSOCK Research Group [<http://research.unir.net>].

References

- [1] J. Carter, M. Markel, Web accessibility for people with disabilities: an introduction for web developers, *Prof. Commun. IEEE Trans.* 44 (2001) 225–233, <http://dx.doi.org/10.1109/47.968105>.
- [2] Web Accessibility Initiative, 2016. <https://www.w3.org/WAI/intro/accessibility.php>.
- [3] European Commission Web Accessibility, 2010 (n.d.) <https://www.w3.org/WAI/intro/accessibility.php>.
- [4] R. Gonçalves, J. Martins, F. Branco, A review on the Portuguese enterprises web accessibility levels – a website accessibility high level improvement proposal, *Procedia Comput. Sci.* 27 (2014) 176–185, <http://dx.doi.org/10.1016/j.procs.2014.02.021>.
- [5] Section 508 Standards Guide, <http://www.section508.gov/section-508-standards-guide2010>.
- [6] P.T. Jaeger, Assessing section 508 compliance on federal e-government web sites: a multi-method, user-centered evaluation of accessibility for persons with disabilities, *Gov. Inf. Q.* 23 (2006) 169–190, <http://dx.doi.org/10.1016/j.giq.2006.03.002>.
- [7] H. Okada, H. Arakawa, T. Kondo, Comparison of web accessibility within Japanese educational institution websites, *ICCAS-SICE*, Vol. 2009 2009, pp. 3605–3608.
- [8] D.A. Bradbard, C. Peters, Y. Caneva, Web accessibility policies at land-grant universities, *Internet High. Educ.* 13 (2010) 258–266, <http://dx.doi.org/10.1016/j.iheduc.2010.05.007>.
- [9] Y. Shi, The accessibility of Chinese local government web sites: an exploratory study, *Gov. Inf. Q.* 24 (2007) 377–403, <http://dx.doi.org/10.1016/j.giq.2006.05.004>.
- [10] M. Reichling, S.S.-S. Cherfi, Integrating accessibility as a quality property in web developments, *Res. Challenges Inf. Sci. (RCIS)*, 2013 IEEE Seventh Int. Conf. 2013, pp. 1–12 doi:10.1109/RCIS.2013.6577698.
- [11] A.L. Dias, R. Pontin de Mattos Fortes, P.C. Masiero, Increasing the quality of web systems: by inserting requirements of accessibility and usability, *Qual. Inf. Commun. Technol. (QUATIC)*, 2012 Eighth Int. Conf. 2012, pp. 224–229 doi:10.1109/QUATIC.2012.33.
- [12] M.-L. Sánchez-Gordón, L. Moreno, Toward an integration of web accessibility into testing processes, *Procedia Comput. Sci.* 27 (2014) 281–291, <http://dx.doi.org/10.1016/j.procs.2014.02.031>.
- [13] Checklist of checkpoints for web content accessibility guidelines, <http://www.w3.org/TR/WCAG10/full-checklist.html>1999.
- [14] A11YProject, 2016. <http://a11yproject.com/checklist.html>.
- [15] V.L. Centeno, C.D. Kloos, J.A. Fisteus, L.A. Álvarez, Web accessibility evaluation tools: a survey and some improvements, *Electron. Notes Theor. Comput. Sci.* 157 (2006) 87–100, <http://dx.doi.org/10.1016/j.entcs.2005.12.048>.
- [16] B.P.-C. Yen, The design and evaluation of accessibility on web navigation, *Decis. Support. Syst.* 42 (2007) 2219–2235, <http://dx.doi.org/10.1016/j.dss.2006.07.002>.
- [17] H. Dworak, Automatic temporal evaluation of the accessibility of the World Wide Web and its standards conformance, *Dependability Comput. Syst.* 2008. DepCos-RELCOMEX '08. Third Int. Conf. 2008, pp. 171–178 doi:10.1109/DepCos-RELCOMEX.2008.27.
- [18] WAVE Web accessibility evaluation tool, 2016. <http://wave.webaim.org>.
- [19] AChecker web accessibility checker, 2016. <http://achecker.ca>.
- [20] L. Moreno, P. Martínez, B. Ruiz, A. Iglesias, Toward an equal opportunity web: applications, standards, and tools that increase accessibility, *Computer* 44 (2011) 18–26, <http://dx.doi.org/10.1109/MC.2010.370>.
- [21] R.G. Crespo, L.J. Aguilar, O.S. Martínez, Improving access to [IT] services for people with disability through software aids, *JAISE* 4 (2012) 563–564 doi:10.3233/AIS-2012-0181.
- [22] J.F. Lima, G.M. Caran, L.F.R. Molinaro, D.F. Garrossini, Analysis of accessibility initiatives applied to the web, *Procedia Technol.* 5 (2012) 319–326, <http://dx.doi.org/10.1016/j.protcy.2012.09.035>.
- [23] G.C. Vanderheiden, Design of HTML (mosaic) pages to increase their accessibility to users with disabilities strategies for today and tomorrow, http://trace.wisc.edu/archive/html_guidelines/version1.html1995.

- [24] R. González Crespo, La Web 3.0 al servicio de las personas discapacitadas auditivas mediante las pautas de accesibilidad 2.0, *Soc. Y Utopía. Rev. Ciencias Soc.* 2010, pp. 153–172.
- [25] N. Fernandes, D. Costa, C. Duarte, L. Carriço, Evaluating the accessibility of web applications, *Procedia Comput. Sci.* 14 (2012) 28–35, <http://dx.doi.org/10.1016/j.procs.2012.10.004>.
- [26] M. Kirchner, Evaluation, repair, and transformation of web pages for web content accessibility. Review of some available tools, *Web Site Evol.* 2002. Proceedings. Fourth Int. Work. 2002, pp. 65–72 doi:10.1109/WSE.2002.1134091.
- [27] 508Checker, 2016. <http://www.508checker.com>.
- [28] Evaluera, 2016. <http://www.evaluera.co.uk>.
- [29] Access Monitor, 2016. <http://www.acessibilidade.gov.pt/accessmonitor/>.
- [30] Accesslint, 2016. <http://accesslint.com>.
- [31] Cynthiasays, 2016. <http://www.cynthiasays.com>.
- [32] Sortsite, 2016. <http://www.powermapper.com/products/sortsite>.
- [33] EvalAccess, 2016. <http://sipt07.si.ehu.es/evalaccess2/>.
- [34] S. Abou-Zahra, A data model to facilitate the automation of web accessibility evaluations, *Electron. Notes Theor. Comput. Sci.* 157 (2006) 3–9, <http://dx.doi.org/10.1016/j.entcs.2005.12.040>.
- [35] V.L. Centeno, C.D. Kloos, M. Gaedke, M. Nussbaumer, WAEX: web accessibility evaluator in a single XSLT file, *Autom. Specif. Verif. Web Syst.* 2006. WWW '06. 2nd Int. Work. 2006, pp. 29–36 doi:10.1109/WWW.2006.14.
- [36] T.-H.-G. Vu, D.T. Tuan, V.-H. Phan, Checking and correcting the source code of web pages for accessibility, *Comput. Commun. Technol. Res. Innov. Vis. Futur. (RIVF)*, 2012 IEEE RIVF Int. Conf. 2012, pp. 1–4 doi:10.1109/rivf.2012.6169872.
- [37] J.J. Mayol, G.F. Nadal, A web accessibility improvement tool, *Inf. Syst. Technol. (CISTI)*, 2011 6th Iber. Conf. 2011, pp. 1–5.
- [38] A. Garrido, S. Firmenich, G. Rossi, J. Grigera, N. Medina-Medina, I. Harari, Personalized web accessibility using client-side refactoring, *Internet Comput. IEEE* 17 (2013) 58–66, <http://dx.doi.org/10.1109/MIC.2012.143>.