GAP4APE: a client-side system to enhance accessibility in the Web 2.0 era

Silvia Mirri  
Department of Computer Science  
University of Bologna  
Via Mura Anteo Zamboni 7  
40127 Bologna (BO), Italy  
silvia.mirri@unibo.it

Paola Salomoni  
Department of Computer Science  
University of Bologna  
Via Mura Anteo Zamboni 7  
40127 Bologna (BO), Italy  
paola.salomoni@unibo.it

Giorgia Cucchiarini  
Corso di Laurea in Scienze dell’Informazione  
University of Bologna  
Via Sacchi, 3  
47521 Cesena (FC), Italy  
giorgia.cucchiarini@studio.unibo.it

ABSTRACT

Social networking systems have radically changed personal communication by providing new and interesting opportunities both in leisure and in business. Despite their apparent universal appeal, these systems are effectively leaving out a part of users with disabilities, who have difficulties in register, join and participate in the main on-line communities. Facebook represents a critical case: beside some accessibility improvement provided by this social network service, its accessibility represents an issue yet. This paper presents an augmented browsing system, which allows users with disabilities to specify their needs and preferences about Web pages presentation, directly by means of the browser interface. On the basis of the declared settings, the system automatically transcodes the page content (both static and dynamic), producing new pages adapted to the user’s needs. The system, based on the widespread Web browser extension called GreaseMonkey, works on a wide set of Web systems, but has been extensively tested on Facebook.

INTRODUCTION

Online social networks gained an exponentially growing number of members by offering users new, effective tools for interaction and communication (Isaías 2009). By using Facebook and other social networking systems, people keep in touch each other, meet new friends and connect again with old ones, find jobs and discover new interests. Moreover, users express themselves by sharing textual contents together with pictures, music and videos.

The lack of accessibility of main social networks systems is a widely known issue since these social network services become so pervasive. During the last years just few and partial solutions to overcome this lack were provided by developers (Hailpern 2009, Zajicek 2007). Recently, Facebook accessibility has been enhanced by removing some technological barrier (a CAPTCHA image) from its sign up interface. Currently Facebook accessibility is mainly addressed in a very simple and incomplete way, substantially based on the use of the mobile version of the system. Analogously Twitter, MySpace and other widely used social networks are partially accessible to users with disabilities (AbilityNet 2008) and are not compliant with any national regulation or W3C guidelines (W3C 2008). This accessibility issue is quite common in most of Web 2.0 services, which are strongly based on smart interfaces implemented with a wide use of AJAX scripts (Garrett 2005). Social networks, and more specifically Facebook, support users’ customization as the possibility of changing few layout characteristics, such as color background or text size. These modifications are bound to the user profile page only and they have to be manually specified by the user without any support by some profiling tool.

In this paper we present GAP4APE (GreaseMonkey And Profiling for Accessible Pages Enhancement), a system to improve page accessibility based on GreaseMonkey augmented browsing (Greaspot 2011). The system works by automatically modifying the page content on the basis of a profile describing the user’s needs and preferences (Mirri et al. 2011). This task is carried out on the basis of a GreaseMonkey through transcoding techniques in order to provide the user an adapted and optimized version of each page, according to his profile. Such modifications are applied both to static content (like text, images or embedded multimedia) and to dynamic content (as AJAX scripts) that controls the interface dynamics. In order to describe user’s needs and preferences in GAP4APE, we have used a profiling system, based on the well-known IMS ACCLIP standard (IMS 2002a). GAP4APE can support user in navigating every site by means of a set of customized client-side transcoding scripts that could be developed by a community of users. GAP4APE activates the appropriate sets of script on each page, if a specific group of transcoding activities is available. Otherwise, a standard set of real-time adaptation mechanisms is applied, thereby improving accessibility of all the sites browsed by the user.

We provide a set of scripts devoted to improve accessibility of Facebook interface as a test suite and case study for the effectiveness of the system. In order to design and develop scripts which improve Facebook accessibility, some people with disabilities have been invited to report how they use their assistive technologies/tools while navigating Facebook. This group of users has been involved during the scripts design phase and also during the testing one.

© EUROSIS-ETI
The reminder of this paper is organized as follows. The next Section ("The GAP4APE System") presents main design issues and the system architecture. Section entitled "Facebook accessibility" describes main accessibility issues of Facebook, while Section entitled "GAP4APE at work" shows how our system works on the case study of Facebook. Finally, "Conclusions and Future Work" Section ends the paper by introducing main conclusion and future work.

**THE GAP4APE SYSTEM**

GAP4APE is a client-side augmented browsing system designed and implemented with the aim of enhancing Web pages accessibility by dynamically and automatically updating them to specific user’s needs. The design of GAP4APE has been driven by the idea that "the best Web content for each one" philosophy should replace the more traditional idea of accessibility claiming that "one Web content for everyone" (Salomoni et al. 2008). On one hand, Web accessibility principles assert that using more than a Web page to provide differentiated content to users with disabilities is intended as a discriminating and segregation factor (Bohman 2003). On the other hand, with the pervasive diffusion of client-side technologies, the customization of content has become a widely use technique to improve usability. Moreover, it is worth noting that a single accessible Web page, providing a unique content, could not be optimized for each user who browses it (Salomoni et al. 2008). Concepts and techniques for content transcoding offer the means of generating optimized versions of a same primary resource, providing each user with content adapted to his/her needs and preferences (Bigham 2007, Bigham and Ladner 2007).

**User profile**

On the basis of the user’s profile, GAP4APE performs a set of transcoding scripts. The system provides the user with a browser interface to set his/her preferences and needs defining his profile. To specify a profile that effectively describes special needs of users with disability (including characteristics and settings of assistive tools) we have used a significant part of a well-known standard, the IMS ACCLIP (Accessibility for Learner Information Package) (IMS 2002a).

IMS ACCLIP is a part of the IMS Learner Information Package (IMS LIP) specification (IMS 2002b). In such a part, a set of packages is defined, which can be used to import/export data into/from an IMS compliant e-learning platform. In particular, in IMS ACCLIP the user is described in terms of accessibility needs by the means of a XML-based syntax. Practically speaking, ACCLIP describes user’s preferences and needs (visual, aural of device), in order to allow a learning content customization on the basis of them (e.g. preferred/required input/output devices and/or preferred/required content alternatives). Thus, such a profile could be exploited as a description of how users interact with an e-learning environment by focusing on accessibility requirements (Mirri et al. 2011). The ACCLIP specification defines the required elements to represent accessibility preferences, which can be grouped into the following sections:

1. **display information**: this section describes how the user prefers to have information displayed or presented;
2. **control information**, this section defines how a user prefers to control the device;
3. **content information**, this section describes what enhanced, alternative or equivalent content the learner requires;
4. **accommodations**, this section allows recording of requests for and authorization of accessibility accommodations for testing or assessment.

GAP4APE profiling considers attributes from the a, b and c section, excluding the ones of the d section. In particular, our profiling system groups the preferences and needs information into Text, Color, Audio, Visual and General sets.

```
...<accessForAll schemaVersion="1.0.29"
xmlns:xsi="http://www.imsglobal.org/xsd/acclip"
xmlns:xsi:schemaLocation="http://www.imsglobal.org/xsd/AccessForAllv1p0.xsd">
<context identifier="userX" xml:lang="it">
  <display>
    <screenReader>...
    </screenReader>
    <screenReaderGeneric>
      <link value="speakLink"/>
      <link value="differentVoice"/>
      <speechRate value="500"/>
      <pitch value="0.8"/>
      <volume value="0.5"/>
    </screenReaderGeneric>
    </display>
    <braille>
    </braille>
    <control>
      <mouseEmulation>...</mouseEmulation>
      <voiceRecognition>...</voiceRecognition>
    </control>
  </context>
</accessForAll>
...
```

Figure 1: Fragment of a profile set by a blind user

Let us consider a blind user who gains access to the Web with a PC equipped with a screen reader and a Braille display (i.e., the assistive technologies that enable blind people to use a computer). A simplified portion of ACCLIP derived elements comprised in the user’s profile is reported in Figure 1. In particular, a set of preferences is specified, related to the use of the screen reader (see element <screenReader> inside <display> element), as well as the Braille display characteristics (see <braille> element, partially omitted). All these elements are included inside the accessibility LIP element (<AccessForAll>) which drives the system transcoding process. The system adequately transcodes the Web pages on the basis of such profile.

© EUROSI-ETI
Transcoding

On the basis of the above cited profile, GAP4APE automatically modifies the page content, by adapting it to the chosen features. This activity is performed by a set of transcoding scripts, selected by the system considering both the user profile and the page content. Recently, augmented browsing technologies offer the opportunity to create effective client-side transcoding applications. We have decided to use GreaseMonkey, a browser extension that allows users to develop, install and use scripts which make on-the-fly changes to HTML DOM page (Greasop 2011, Pilgrim 2005). Since its first release, GreaseMonkey was used to support Web users in automatically adapting Web pages for accessibility purposes. Currently GreaseMonkey is also supported by some mobile devices (Greasop 2011).

In order to enhance the accessibility of Web content and to provide the best adaptation to each user, by meeting his/her needs and preferences, our scripts allow the transcoding of Web pages by modifying the CSS rules, the HTML DOM and also Web 2.0 scripts. Some examples of such adaptations applied by our scripts will be described in Section “GAP4APE at work”.

The system recognizes the Web application browsed by the user and verifies the availability of specialized scripts to perform a page-specific adaptation. If such a set of script is not available, GAP4APE applies a general set of default scripts to improve the page accessibility. Our system adaptation mechanisms works in analogy with well-known screen readers (i.e. Jaws (Freedom Scientific 2011)). These assistive tools act on desktop applications applying customized scripts, whenever they are available and default scripts otherwise. It is worth noting that screen reader users frequently write their own scripts to improve accessibility of applications and share them with other users. Analogously, it is possible for GAP4APE users to write specific adaptation scripts to be shared with other users, as in a crowdsourcing community.

Architecture

The following Figure 2 shows GAP4APE architecture: each user is provided (at the client-side) with a browser operating the GreaseMonkey extension.

GAP4APE works as a set of GreaseMonkey scripts, structured into two modules: ProM, Profiling Module, devoted to store and manage the user profile, and TraM, content Transcoding Module that performs the adaptation of content.

Once the page from the server has been downloaded, GAP4APE TraM verifies the set of scripts to be applied (a specific one or the more general default scripts set) and matches it with the profile defined by the user. The new version of the page, adapted by considering the user’s needs and preferences, is displayed to the user. The profile can be edited and managed through the GAP4APE ProM directly by the user through the browser interface.

Figure 2: GAP4APE Architecture

Figure 2 shows how three users, once they set their different profiles, browse three different Facebook pages, derived from the same original one. More specifically:

- User A, with low vision, receives a high contrast page, adapted to be browsed by a screen magnifier user.
- User B, who is blind, receives a page with a simplified and linearized structure, adapted to be browsed by a screen reader user.
- User C, as an empty profile, receives the Facebook page as it is, without any adaptation.

Details on adaptation performed on Facebook page are reported in the following Section “GAP4APE at work”.

A GAP4APE prototype has been developed and it has been tested by a group of users with visual impairments (blind people and people with low vision). The entire system will be completed as a set of different extensions tailored for different Web browsers. In this way, GAP4APE will improve Web pages accessibility independently from the browser the user exploits to navigate. The current prototype of GAP4APE has been implemented for Mozilla Firefox, on the basis of the GreaseMonkey mechanism which is devoted to build extensions (Greasop 2011).

FACEBOOK ACCESSIBILITY

Since its birth Facebook has shown several accessibility problems, which have been mainly due to the heavy use of Web 2.0 and scripting technologies. In particular, a study which was conducted in 2008 on main online social network accessibility (AbilityNet 2008) has rated Facebook “very inaccessible” and it has shown that main Facebook accessibility lacks are related to the following issues:

a. Facebook required the user to identify a CAPTCHA image when creating an account. This was a barrier for vision impaired users, particularly screen reader users, but also users with minor vision impairments or dyslexia.

b. There was no separate accessibility page on any of the websites tested. Many websites now display an accessibility page as a way of making a public statement of commitment to ensuring disabled access to their website and to explain any measures,
such as access keys which allow keyboard users quick access to important pages. In addition, an accessibility page can be used to provide extra information for disabled users, such as contact details for the organization in question.

c. Facebook requires JavaScript to be enabled otherwise a user cannot create an account. Scripts can often cause difficulties for those using older browsers, those with vision impairments using some special browsers, and those whose organizations disable JavaScript for security reasons.

d. If an existing Facebook user logs into the website, yet has JavaScript disabled, there will be various functionalities unavailable to them such as drop down menus not working and the ability to set your phone up to send photos directly to your Facebook account.

e. Text size on every page cannot easily be resized by many users – so vital for many visitors who have vision impairment or who are viewing the website on a small screen. A user should override the default text size by changing their browser settings, some website content can overlap and make the text difficult or impossible to read.

f. Users of screen reading software pull all links on a page into a list – so they can quickly access the link they want. Many links do not make sense when read out of context in this way, such as many occurrences of “See all” or “Change” that each takes the user to a different page, or perform a different action.

g. Keyboard only users would experience varying degrees of difficulty, ranging from a lack of links that allow them to jump over main navigation links, to pages or features that were effectively inaccessible to keyboard users.

h. Many graphics lack any alternative text (or “tool tips”). This is true of both graphics that are essential to navigation as well as graphics used purely for cosmetic purposes, causing problems especially for blind screen reader users and that using voice recognition.

i. Although Facebook offered the option to users of linking and including videos, it does not mention the importance of captions (or subtitles), and it does not give the option of adding a transcript to a video.

Even if in the last years Facebook developers have worked to solve these issues, some of the main accessibility problems remain. In particular, the CAPTCHA image is currently shown together with an auditory one, so that users with visual impairments can sign up to the system. Moreover, Facebook added a page to present its main accessibility features.

A discussion group of people with disabilities has been involved, answering to some interviews and participating in the design. Another group of users has been involved in the testing phases. It is worth noting that each user has got a proper way to navigate the Web and to use his/her own assistive technology. Let us take into account blind users: they subjectively enjoy the features of screen readers, depending on their experiences, their skills, their knowledge about such tools and the frequency they use them.

From a sample of 16 blind users who have been interviewed about how they navigate the Web through a screen reader, we know that all of them use the combination of TAB key and arrows keys, only six of them are used to search text by using the combination of CTRL key and F key, only one of them uses the combination of INS and F6 keys to obtain a list of the headings in the Web page and, finally, 10 of them use the combination of INS and F7 keys to obtain a list of all the links in the Web page.

In general, blind users who navigate Facebook pages through a screen reader face different levels of barriers and meet different problems. Some of these main problems can be summarized as follows:

- The chat is not accessible.
- Headings are not well-organized and their hierarchy is not clear.
- Some links provide a cyclic navigation, without a clear destination.
- Some important and useful features and parts of the content are difficult to be reached.
- Useless information and images makes the navigation difficult and heavy.
- Some text links are ambiguous.
- Some links and some information are redundant.
- Some useful features are read as simple text instead of button titles, links or labels (e.g. the “Comment” feature).
- There are some difficulties in finding friends when coincidences of names happen.
- Each update refreshes the whole page.

A similar set of issues has been identified by studying the interaction with Facebook of a group of users with low vision and users with color blindness.

**GAP4APE AT WORK**

This section is devoted to describe the Facebook use case. We will illustrate how a blind user and a low vision user can set their preferences and how they can enjoy the social network.
Setting preferences

The user can improve the navigation experience by defining his/her needs and preferences through the Preferences Panel (PP). In the current version of GAP4APE prototype, such a Panel is available as a menu choice in the Firefox interface and it is provided to users as a window which is displayed over the browser one.

The Panel window has been created by using XUL (XML User Interface Language) (Mozilla 2011), which is an XML user interface markup language developed by the Mozilla Project. XUL is a Mozilla’s XML-based language that permits to build feature-rich cross platform applications. Such applications can be customized with different text, graphics and layout. XUL applications are based on other W3C standard technologies too, such as HTML 4.0, CSS 1 and 2, DOM Levels 1 and 2, JavaScript 1.5, ECMA-262 Edition 3 (ECMAScript). Plus, XUL takes into account also the W3C eXtensible Bindings Language (XBL), which is a markup language to define special new elements, or “bindings” for XUL widgets. XBL allows developers to extend XUL by tailoring existing tags and creating new ones. Thus, developers can define and create customized user interface widgets. XUL provides a clear separation among the client application and programmatic logic (XUL, XBL and JavaScript), presentational aspects (CSS and images) and language-specific text labels (DTDs). This way, the layout and appearance of XUL applications are independent from the application definition and logic. This enabled us to define a Preferences Panel which is accessible itself.

The PP organizes all the configurable characteristics into the following sets: Text, Color, Audio, Visual and General. Figure 3 and Figure 4 show screenshots taken from the Color and the Visual Tabs of the Preferences Panel.

Adapting Facebook

The GAP4APE content transcoding system is based on a set of scripts which have been developed by using JavaScript. Such scripts interact with the Web page DOM and they add, remove or change elements, on the basis of users’ preferences and needs, as they are declared into the profiles. For instance, it is possible to remove links or images or any other kind of visual elements such as advertisements, to change the links text, to show the alternative descriptions instead of the related images or to add them and show them close to the related images. Moreover, changes to the CSS are applied by using JavaScript too. Some scripts can directly modify single CSS rules, while some others can create a new CSS file, by applying the user’s preferences and needs, and then they substitute the old CSS with the new one in the HTML code of the Web page. With this kind of scripts it is possible to modify layout and presentational aspects, such as the colors used in the Web page or the font family. Finally, also script transcoding is realized by using JavaScript. Some of these scripts are developed to automatically modify Web 2.0 scripts (for instance to avoid automatic refreshing and/or updating of the Web page), while some others are designed to be executed only on a proper Web application, since it is not always possible to automatically identify and modify AJAX scripts in a feasible and effective way. Hence our prototype applies suitable scripts to transcode specific Web application pages (in particular when the users request Web 2.0 social networks content and services), in a way which is similar to screen readers behavior with desktop applications.

In order to increase Facebook accessibility we have designed and developed a set of scripts which face the problems
described in section “FACEBOOK ACCESSIBILITY”. In particular, our scripts:
• Re-label text links in order to avoid ambiguous links.
• Remove redundant links and information (in particular into Users’ profile pages).
• Label form elements.
• Remove useless images from Users’ profile pages and from the Wall.
• Provide a more accessible chat.
• Block the automatic updating and allow users to choose when refreshing the page.
• Assign and reorganize the headings hierarchy.
• Reorganize lists and nested list items.
• Reorganize the whole layout of the page, grouping in a fixed area all the advertisements and all the information which makes uneasy the navigation with a screen reader.

When a blind user declares his/her preferences and needs by the means of the GAP4APE Preferences Panel, he/she could choose to substitute images with textual alternatives or the preferred kind of alternative (textual, auditory, etc.). Figure 5 shows a screenshot of the original Facebook wall, while Figure 6 represents the same wall with the application of GAP4APE scripts.

Now let us take into account users with low vision. They face different problems; in particular, it is very difficult to meet their needs, since there are several kinds and levels of these visual impairments. The Preferences Panel allows users to set a wide group of settings in a very detailed manner, so as to better transcode the Web pages and to better meet users’ needs and preferences.

An accessible version of Facebook User’s Profile page is depicted in Figure 8: through our GAP4APE scripts the user has set different text and background colors with a high contrast and a bigger text size. In fact, the original color contrast is not enough for user with low vision, moreover...
black background and yellow texts provides a good level of contrast.

Finally, Figure 9 shows another accessible version of the same Facebook User’s Profile depicted in Figure 7 and Figure 8. In this case the applied scripts increase the text size, changing only the color of links (in order to make them more visible) and they reorganize the Web page layout so as to support users with low vision by the means of a simpler and easier to navigate interface.

In order to evaluate our GAP4APE prototype we have conducted a testing phase. We have involved 10 users with visual impairments and asked them to complete a set of task on Facebook pages (such as: to comment a specific post on the wall, to modify profile and privacy settings, to remove a post from the profile page, etc.) by the means of GAP4APE scripts. After that they evaluation testers have answered to a questionnaire. According to the questionnaire answers all the users have completed all the tasks, 70% of testers have found easier to comment and remove posts, while 30% did not found differences between the original version of Facebook and the customized one, 80% of tester prefer transcoded Facebook pages in order to modify profile and privacy configurations and all the users have found easier the whole interface of Facebook wall and profiles pages.

CONCLUSIONS AND FUTURE WORK

In this paper we have presented GAP4APE, a client-side transcoding system with the aim of adapting Web pages in order to enhance Web accessibility, by meeting user’s preferences and needs.

To reach this goal, we have faced two different issues:

1. Profiling users’ preferences and needs. Dealing with users with disabilities we have defined a profiling system which is based on the well-known IMS ACCLIP standard.

2. Transcoding Web pages. We have exploited a two-layers system which applies: (i) a specific set of scripts devoted to a given Web application when such scripts are available; (ii) a default set of scripts otherwise. Transcoding activities are performed on the client-side: a Web page is delivered to any user, but GAP4APE adapts it by transcoding Web content, CSS rules, the HTML DOM and also Web pages scripts. The two-layer transcoding system answers to the need of automatically identifying and modifying AJAX scripts. This kind of adaptation has to be performed by sets of scripts, specifically designed for a given application (see 2.i). Default scripts are not effective in adapting code in a unique and feasible way. The idea of ad hoc scripts for specific applications has been inspired by the behavior of well-known screen readers, such as Jaws.

This paper illustrates a use case of GAP4APE in improving the accessibility of one of the most common and widely used Web 2.0 social network: Facebook. In order to design and implement scripts which enhance Facebook accessibility, it has been analyzed how people with disabilities exploit their assistive technologies while they navigate Facebook. This group of users has been involved during the design phase and also during the testing one.

Future works will be mainly addressed to the integration of our system into a wider set of browsers (including Chrome) and to the definition of Web services which provide automatic content transcoding, involving also multimedia ones, in order to overcome JavaScript limits in providing complex content transformation. Moreover, specific scripts will be designed and developed in order to improve other Social Network applications, such as Twitter and LinkedIn.

ACKNOWLEDGEMENTS

The authors wish to thank Ludovico Antonio Muratori and Catia Prandi for their precious support.

REFERENCES


AUTHOR BIOGRAPHY

SILVIA MIRRI is an assistant professor of Computer Science at the Department of Computer Science of the University of Bologna. Her research interests include: multimodal interaction, accessibility and e-learning.

PAOLA SALOMONI is an associate professor of Computer Science at the Department of Computer Science of the University of Bologna. Her research interests include the following: distributed multimedia systems, wireless multimedia, e-learning accessibility.

GIORGIA CUCCHIARINI is a computer science undergraduate student. She is working on her degree thesis which concerns Web 2.0 social networks accessibility.