Augment browsing and standard profiling for enhancing Web accessibility

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.salomini@unibo.it

Catia Prandi
Corso di Laurea in Informatica
University of Bologna
Via Mura Anteo Zamboni 7
40127 Bologna (BO), Italy
catia.prandi@studio.unibo.it

ABSTRACT

The opportunity of effectively tailoring Web resources presentation - depending on each single user needs and preferences - represents a challenge and a necessity for accessibility and inclusion. On the Web, customizing means transcoding content according to some user and/or device (contextual) settings. Such a profiling refers to devices constraints, user habits, skills, different needs (or tastes) about interaction, in order to drive all the necessary procedures for content (re)shaping. The usual set up that users provide for assistive tools such as screen readers or speech-to-text applications, is a common practice (and a typical example) for a subjective, better enjoyment of resources.

This work describes an augment browsing system, which allows users to set up their needs and preferences about Web pages presentation from the browser interface and is capable to automatically modify (transcode) content, according to such settings at client-side. The system is based on a widespread Web browser extension (GreaseMonkey) and well-known standards have been utilized to represent user's settings. Finally a case-study of the system has been assessed on a widespread social network, also taking into account some evaluations about accessibility by a group of blind persons.

Categories and Subject Descriptors

H.5.1 [**Hypertext/Hypermedia**]: User issues; H.5.2 [**User Interfaces**]: User-centered design; K.4.2 [**Social Issues**]: Assistive technologies for persons with disabilities.

General Terms

Design, Human Factors.

Keywords

Web Accessibility, Web 2.0, Web browser extensions, Augment Browsing, Users Profiling.

1. INTRODUCTION

During the last years, we have witnessed the birth and triumph of Web 2.0 technologies and philosophies. They have pushed the Web onto a new evolutionary stage, by promoting users' role, by greatly increasing the interaction among them and with the Web, as well as by allowing them to easily create and share content on the Net [15].

From a technological point of view, the novelty of Web 2.0 is the introduction (or rediscover) of AJAX (Asynchronous JavaScript and XML) as a client-side development tool/technique for Web applications. Basically, AJAX programming uses JavaScript to upload and download new data from the Web server without reloading the entire page [9]. Indeed, as it concerns to the server side, Web 2.0 uses the same technologies as Web 1.0 did, without introducing any significant novelty.

As it has often happened, the introduction of newer and more interactive technologies has affected accessibility for users who enjoy the Web by using assistive technologies [29], [11]. In particular, the partial download of new data, the continuous refreshing and the massive use of scripting represent significant barriers for those users who navigate the Web by means of a screen reader.

Besides content dynamics and high interaction, Web 2.0 applications have shown a more and more massive presence of multimedia (or rich media) content. Once again, this aspect has deeply affected Web accessibility and it fed a plethora of research projects about content transcoding and adaptation systems. Despite the plenty of efforts, such systems [6], [17], reveal limitations related to several aspects, such as the necessity of a pre-defined set of available content (which could not meet all the users' needs [6], [17]), the lack of a profile which clearly describes users' needs and preferences, the architectural-dependent approach (e.g. a host machine is needed [6], [17]) or the functional-dependent one (e.g. a specific platform must be used).

In this scenario, our idea is to exploit a Web 2.0 characteristic (which seems to be a weakness in terms of Web accessibility) as a tool for enhancing it: using scripts at client-side to provide a content transcoding system which adapts Web pages in order to meet each user's preferences and needs.

In this paper we present GAPforAPE (GreaseMonkey And Profiling for Accessible Pages Enhancement), a system which is based on a set of Web browser extensions with the aim of improving Web page accessibility. GAPforAPE has been designed with the idea in mind that "*one Web content for everyone*" is not as effective as "*the best Web content for each one*" [23]. Concepts and techniques of content transcoding and new Web 2.0 technologies has introduced the opportunity of providing the same Web page to any user, but in an adapted and optimized version, according to some user's profile in terms of needs and preferences [2]. Thus, we have equipped GAPforAPE with a user profiling system, based on the well-known IMS ACCLIP standard [13].

The main idea is modifying Web content by using client-side scripting languages, the well-known GreaseMonkey augment browsing [10] and the AccessMonkey Framework [2] [3]. Our system has been designed in order to be a set of Web browser extensions which includes a profiling system and a client side content transcoding one, based on a collection of scripts. 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 other scripts which are used on them.

One of the notable differences among our proposed system and other well-known Web browser extensions, such as GreaseMonkey and AccessMonkey, is the use of a profiling system, in order to better describe users' preferences and needs. Moreover, since it is not possible to automatically identify and modify AJAX scripts in a feasible and effective way, we have designed our system so that is capable to recognize the Web application and to apply the adequate scripts transcoding, just like well-known screen readers (i.e. Jaws [16]) acts with desktop applications. This permits our system to effectively improve the accessibility of Web applications which are strongly based on Web 2.0 technologies, such as Facebook and other famous and widely used social networks. To reach this goal the main and most common social networks have been analyzed and some scripts have been designed and developed to provide suitable adaptations as it is described in the following Sections.

In particular, our system applies a specific set of scripts devoted to a given Web application, when such scripts are available, and a default set of scripts otherwise. Transcoding activities are performed on the client-side: a Web page is delivered to any user, but our system adapts it (by transcoding Web content, CSS rules, and scripts). Thus the transcoding system can answer to the need of automatically identifying and adapting AJAX scripts, on the contrary of the default scripts set.

This paper presents a use case to enhance the accessibility of one of the most common and widely used Web 2.0 social network: Facebook [8]. In order to design and develop scripts which improve its 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.

The reminder of this paper is organized as follows. In Section 2 we outline the main background related to the use of GreaseMonkey to improve the accessibility and Section 3 discusses on main design issues and system architecture. Section 4 presents our prototype implementation and Section 5 shows a use case of our system, applied to one of the most common and widely used Web 2.0 social network: Facebook. Section 6 proposes a comparison between our system and some works with similar aims. Finally, Section 7 concludes the paper.

2. BACKGROUND

Research projects in the field of content transcoding and user profiling have not really fallen out on augment browsing. The most part of scripts based on GreaseMonkey and other similar browser extensions do not apply any advanced mechanism for content adaptations.

The use of GreaseMonkey to improve Web content accessibility is quite common and several scripts are devoted to this goal as reported in the following Subsection 2.1. This subsection presents also AccessMonkey, a common scripting framework that web users and developers can use to collaboratively improve accessibility. This project represents one of the most significant research activity on accessibility based on client-side on-the-flight adaptation. Subsection 2.2 briefly introduces accessibility issues in main social networks and describes some basic solutions provided to overcome them. Finally this subsection presents the well-known research and some application issues on user profiling.

2.1 GreaseMonkey and Accessibility

GreaseMonkey is a browser extension that allows users to install scripts which make on-the-fly changes to HTML web page content [10], [22]. Since the first release, GreaseMonkey was used to support web users in automatically transcode web pages for accessibility purposes. Many scripts are available to improve accessibility features of pages, through different script repositories [5], [26]. Main solutions provided through GreaseMonkey scripts for accessibility issues are:

- Alteration of pages that improve accessibility such as adding headers or removing conflicting among keyboard shortcuts.
- Adaptations useful to specific subsets of site users, i.e. augmenting contrast of colors or enlarging fonts for people with low vision. Most part of the scripts are devoted to support visitors of the site who use a screen reader or more generally have a visual disability.
- Improvement or solutions for accessibility that Web developers have not included, e.g. the support for access keys or proper table annotation.
- Overcoming significant barriers to accessibility as, for example, trying to solve or remove CAPTCHAs.

Many of these scripts are used to solve single issues on specific web pages or services. They are organized in repositories not necessarily limited to accessibility purposes, but they are not structured as a whole corpus of code.

To overcome the lack of portability of this approach and enhance the integration among different scripts, in [2] Bigham and Ladner define and implement a framework, called AccessMonkey, devoted to support the development of accessible GreaseMonkey scripts. Through the AccessMonkey Framework, users can edit web pages using JavaScript. This framework is derived by the GreaseMonkey Firefox extension [10] that allows users to inject their own scripts into arbitrary web pages and these scripts can then alter Web pages automatically. AccessMonkey natively supports web accessibility by providing Web developers with appropriate mechanism for editing, approving and saving changes that have been made to web pages by user scripts [3]. The system offers some browser plugins to automatically retrieve user scripts from a common repository and apply helpful transformations to the pages visited by users, in particular:

- the end user interface allows both users and developers to create and share new improvements.
- The developer interface is used by content creators to edit and save changes the scripts made to Web content.

2.2 Social Networks accessibility and Profiling

Accessibility limitations of main social networks is a widely known issue that found just few and partial solutions in the last years [1]. For example, Facebook accessibility is addressed in a very simple and partial way, mainly on the mobile version of the system. Analogously Twitter, MySpace and other widely used social networks are not accessible in conformance to any national regulation or W3C guidelines and present significant accessibility barriers. An accessible version of Twitter was implemented with the idea to offer an alternative access to the platform [18] Accessible Twitter uses a consistent layout for simple navigation in which all text features and color contrast options are optimized for screen readers and audio cues alert the user when nearing the character limit [18]. Advanced issues like the management for content volume and content update or improvements on script dynamics are far from being solved.

Social networks provide users' customization as the possibility of changing few layout characteristics, such as color background or text size. Usually, these modifications are bound to the only user profile page and they have to be manually specified by the user without any support by some profiling tool.

Equipping these platforms with adaptation systems (which were capable to answer any standard profile description) would actually be helpful for accessibility improvements. Profiling user's preferences and needs, as well as device characteristics is a common and useful feature in projects devoted to content adaptation. There are several, different standards which can be used to describe both these sets of capabilities.

Different standardized methods are devoted to profile devices. The most prominent ones are mainly based on RDF (Resource Description Framework) profiles, such as the Composite Capabilities/Preference Profile (CC/PP) [27] and User Agent Profile (UAProf) [21]. These are two related standards, recommended by the W3C and the Open Mobile Alliance (OMA). As the diversity of devices increases, device capabilities and preferences for profiling devices must be known. The goal of these profiles is to allow client devices to inform servers of their capabilities. The CC/PP and UAProf data formats exploit two-level hierarchies, consisting of components and attributes. CC/PP and UAProf are also useful for device independence, content negotiation and adaptation, as they allow different devices to specify their capabilities in a uniform way. CC/PP provides a standard way for devices to transmit their profiles when requesting Web content. Servers and proxies can then provide an adapted content, which is appropriate to that particular device [27]. A CC/PP vocabulary is defined by using RDF and specifies components and their attributes, to be used by the application to describe a certain context. Three main components specify the hardware platform (which describes hardware characteristics, such as display dimensions, boards presence, media support, keyboard type, etc.), software platform (which describes software capabilities, such as operating system name, version, tools and players support, etc.) and browser user agent (which describes capabilities such as user agent name, version, scripting support, CSS support, etc.). UAProf is defined as a standard between WAP devices and servers. The profile can be used for better content adaptation for different types of WAP devices [21]. Some projects have been involved CC/PP and UAProf standards in order to profile devices. The Sun Microsystems Inc. has defined a specification which details a set of APIs for processing CC/PP information, in order to enable interoperability between Web servers and access mechanism, facilitating device independent web applications development [25]. DELI [7] is an open-source library developed at HP Labs, which provides an API to allow Java servlets to determine the delivery context of a client device using CC/PP or UAProf. The Java servlets resolve HTTP requests containing delivery context information form CC/PP or UAProf capable devices and query the resolved profile, replacing proprietary delivery context descriptions with standardized CC/PP descriptions, if it is necessary.

One of the main standard devoted to profile users' accessibility constrains is the IMS ACCessibility for Learner Information Package (ACCLIP) [13]. IMS ACCLIP is a specific part of the IMS Learner Information Package (IMS LIP) specification [14], which has been defined with the aim of addressing interoperability issues among Internet-based learner information systems. The intent of this specification is to define a set of packages that can be used to import (extract) data into (from) an IMS compliant e-learning platform. ACCLIP describes the user in terms of accessibility needs by using a XML-based syntax. Basically, it enables the description of user preferences (visual, aural or device), which can be usefully exploited for tailoring learning contents (e.g., preferred/required input/output devices or preferred content alternatives). In other words, such a personal profile provides a means to describe how users interact with an e-learning environment, by focusing on accessibility requirements.

The ACCLIP specification defines the required elements to represent accessibility preferences, which can be grouped into the following sections:

- *display information*, which describe how the user prefers to have information displayed or presented; for example, it is possible to define preferences related to cursor, fonts and colors characteristics. In addition, it is possible to declare the need of using a screen reader, specifying the interaction preferences, such as the speech rate, its pitch and is volume, or the need of visual alerts instead of aural ones;
- *control information*, which define how a user prefers to control the device; for example, it is possible to define preferences related to standard keyboard usage. In addition, it is possible to declare the need of using non typical control mechanisms, such as an onscreen keyboard, an alternative keyboard, any mouse emulation, an alternative pointing mechanism and any voice recognition;
- *content information*, which describe what enhanced, alternative or equivalent content the learner requires; for example, it is possible to define how to present visual, textual and auditory contents in different modalities and the need of personal style sheets;

- *accommodations*, which allow recording of requests for and authorization of accessibility accommodations for testing or assessment; for example, it is possible to declare the request for accommodations and the accommodation description.

IMS ACCLIP standard is adopted by several and different projects with the aim of profiling accessible users' needs and preferences, as described in [24], [20] and [19].

3. DESIGN ISSUES AND SYSTEM ARCHITECTURE

The main goal of GAPforApe is enhancing accessibility of Web pages by dynamically and automatically modifying them at the client side. Parameters about how and where pages will be altered before being presented are based on the user's profile, which is set up as a set of preferences/needs through a suitable interface of the browser.

The design of GAPforApe has been driven by the idea that "*one Web content for everyone*" is not the same, or is not as effective as "*the best Web content for each one*" [23]. Since the early birth of Web accessibility principles, using parallel Web pages (often created as text-only content) has been intended as a discriminating and segregation factor for people with disabilities [4]. On the other hand, it is worth noting that a unique accessible Web page offers an accessible content which could not be the best one for each user [23]. Concepts and techniques for content transcoding have introduced the opportunity of providing the same Web page to any user in an adapted and optimized version, thereby meeting subjective user's needs and preferences in a suitable way [2].

Besides such opportunities, assistive technologies can generally be configured to tailor content access and navigation to the user's experience, skill, knowledge or simply his/her taste. Content transcoding and assistive tools customization imply any form of profiling (to report user's preferences and device constraints), by typically using attributes and corresponding values. In our system, we have taken into account such an essential common feature, by adopting a significant part of a well-known standard, the IMS ACCLIP (Accessibility for Learner Information Package [13]). As described in the previous Section, IMS ACCLIP is a part of IMS LIP [14] and it has been originally devoted to describe learners' accessibility constraints [12]. Hence, ACCLIP describes the user in terms of accessibility needs, without considering the device characteristics. In particular, ACCLIP enables the description of user preferences (visual, aural or device) that can be exploited for tailoring content (e.g. preferred/required input/output devices or preferred content alternatives).

In our system we have taken into account only attributes belonging to the *display information*, *the control information* and the *content information* sections. In particular, our profiling system groups the preferences and needs information into Text, Color, Audio, Visual and General sets.

Once users have set up their profile, GAPforAPE is capable to modify contents, by adapting them to the chosen features. Indeed, let us state that the core feature of our system is content transcoding. In general, four categories should be mentioned, which represent the most significant, proposed solutions for content transcoding [6], [17], i.e.:

1. *client-side approach*: the transcoding process is in charge of the client application. Client-side solutions can be classified into two main categories with different behaviors: (i) the clients receive multiple formats and adapt them by selecting the most appropriate one to play-out, or (ii) the clients compute an optimized version from a standard one. This approach suggests a distributed solution for managing heterogeneity, supposing that all the clients can locally decide and employ the most appropriate adaptation to them;
2. *server-side approach*: the server (that provides contents) performs the additional functions of content adaptation. In such an approach, content adaptation can be carried out in an off-line or on-the-fly fashion. In the former, content transcoding is performed whenever the resource is created (or uploaded on the server) and a human designer is usually involved to hand-tailor the contents to different specific profiles. Multiple formats of the same resources are thus stored on the server and they are dynamically selected to match client specifications. In all the on-the-fly solutions, adapted contents are dynamically produced before delivering them to the clients;
3. *proxy-based approach*: the adaptation process is carried out by a node (i.e. the proxy) placed between the server and the client. In essence, the proxy captures replies by the server to the client requests and performs three main actions: (i) it decides whether performance enhancements are needed; (ii) it performs content adaptations; (iii) it sends the adapted contents to the client. To accomplish this task as a whole, the proxy must know the target device, the user capabilities (this information must be received from the client) and a "full" version of the original contents (this data must be received from the server). As a consequence, the use of network bandwidth could be intensive in the network link between the proxy and the server;
4. *service-oriented approach*: the dynamic nature of adaptation mechanisms together with opportunities offered by the Web Service technologies, provide an approach of service-oriented content adaptation. The philosophy at the basis of this approach is fundamentally different from those ones previously described, since the transcoding and the adaptation activities are organized according to a service-oriented architecture. Indeed, the number of content adaptation typologies, as well as the set of multiple formats and related conversion schemes is still increasing. This dynamism is one of the reasons that makes difficult developing a single adaptation system that can accommodate all the types of adaptations; therefore, third-party adaptation services are important.

The opportunities of directly operating changes and adding extensions to traditional Web browsers pushed us in taking into account a client-side transcoding system.

The main idea is to modify Web content by using client-side scripting languages, like the well-known GreaseMonkey extension [10] and the AccessMonkey Framework [2], [3]. Thus, our system has been designed in order to be a set of Web browser extensions which includes a profiling system and a client-side content transcoding one, based on a collection of scripts.

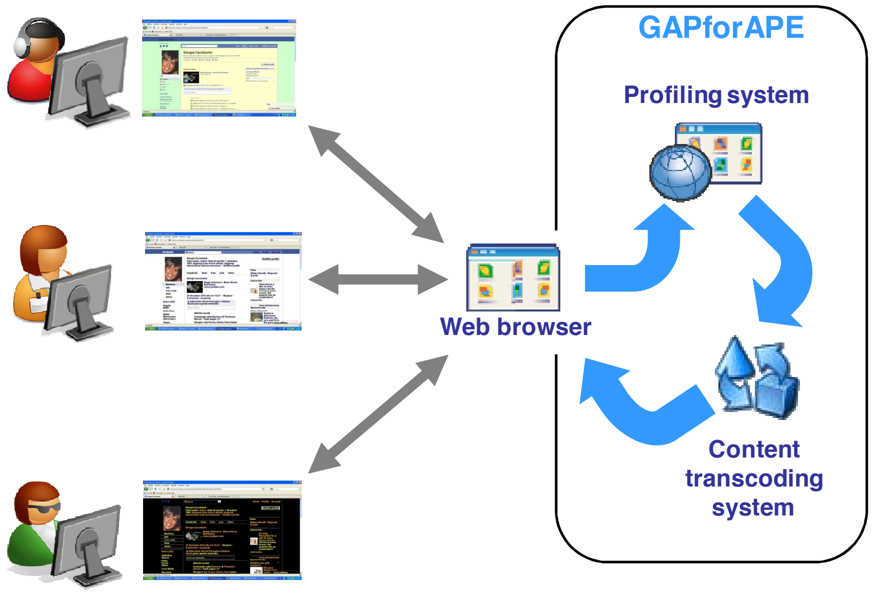


Figure 1. GAPforAPE architecture.

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. Modifications applied by our scripts will be described more in details in the following Section 4.

Figure 1 depicts the architecture of our system, which is mainly based on a Web browser extension, composed by a profiling system and content transcoding system.

The main differences among our proposed system and other well-known Web browser extensions, such as GreaseMonkey and AccessMonkey are the use of a profiling system (in order to better describe users' preferences and needs) and the modification of Web pages script. This permits our system to improve the accessibility of Web applications which are strongly based on Web 2.0 technologies, such as Facebook and other famous and widely used social networks. The use of Web 2.0 technologies really affects the accessibility of such Web applications, as it is illustrated in [29] and [11].

It is worth noting that it is not always possible to automatically recognize and adapt any AJAX script in a feasible and effective way. Hence, we have designed our system so as to identify the Web application and to transcode it by using a specific set of scripts. This behavior has been inspired by the way the well-known screen readers (i.e. Jaws [16]) acts with desktop applications. In particular, we have exploited a two-layer system which applies a specific set of scripts which are devoted to a given Web application (when such scripts are available), otherwise a default set of scripts is used. Hence, a Web page is delivered to any user, but our GAPforAPE adapts it on the client-side (by applying the above mentioned transcoding activities, e.g. transcoding HTML code, CSS rules, scripts). The two-layer solution allows the transcoding system answering to the need of automatically identifying and adapting AJAX scripts, on the contrary of the default scripts set.

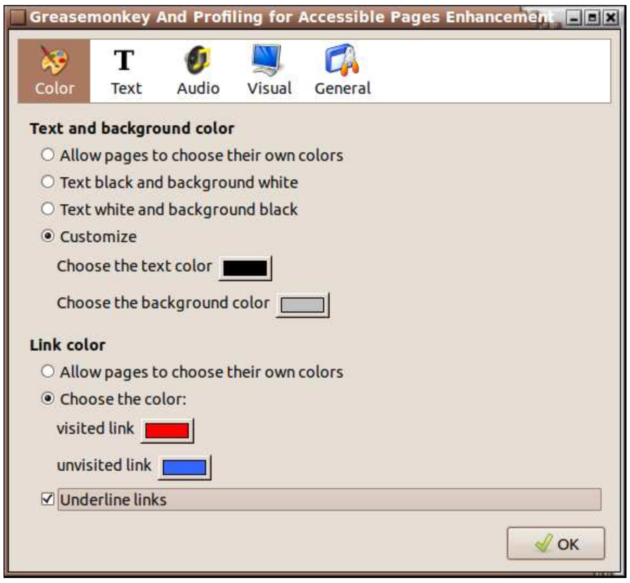
In this way, our proposed system could provide benefits to user with disabilities in enjoying any Web content, including also Web 2.0 social networks. To reach this goal we have studied the main and most common social networks and we have designed and developed sets of scripts to enhance the accessibility of such Web 2.0 applications.

4. IMPLEMENTATION

Currently, a prototype of our GAPforAPE system has been implemented and it has been tested by a group of users with disabilities. The whole system will be implemented as a set of Web browser extensions, so as to enhance the accessibility of Web pages each user navigates.

Our first extension has been developed for Mozilla Firefox, on the basis of the same mechanism of the GreaseMonkey extension [10]. The user can improve his/her navigation by defining his/her needs and preferences through the Preferences Panel. 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. Such a window has been created by using XUL (XML User Interface Language) [28], an XML user interface markup language developed by the Mozilla Project. XUL is a Mozilla's XML-based language that allows building feature-rich cross platform applications which can be customized with different text, graphics and layout. Applications written in XUL are also based on other W3C standard technologies, such as HTML 4.0, CSS 1 and 2, DOM Levels 1 and 2, JavaScript 1.5, including ECMA-262 Edition 3 (ECMAScript). Moreover, XUL takes into account also the W3C eXtensible Bindings Language (XBL), a markup language which defines special new elements, or "bindings" for XUL widgets. XBL enables developers to extend XUL by customizing existing tags and creating new ones. This way, developers can create tailored user interface widgets. One of the main advantages of XUL is its provision of a clear separation among the client application and programmatic logic (consisting of XUL, XBL and JavaScript), presentational aspects (consisting of CSS and images) and language-specific text labels (consisting of DTDs). Hence, the layout and appearance of XUL applications are independent from the application definition and logic. This allows us to create a Preferences Panel (PP) which is accessible itself. The PP organizes all the configurable characteristics into the following sets: Text, Color, Audio, Visual and General.

Figure 2 and Figure 3 show screenshots taken from the Color and the Visual Tabs of the Preferences Panel.



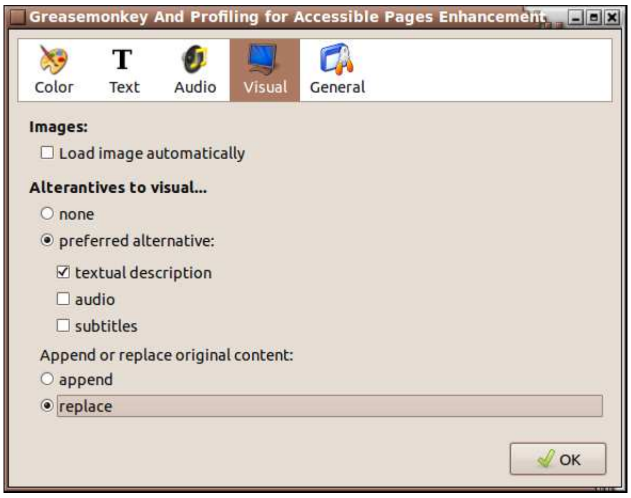
Text and background color

- Allow pages to choose their own colors
- Text black and background white
- Text white and background black
- [✓ Customize]
  - Choose the text color ⬛
  - Choose the background color ⬜

Link color

- Allow pages to choose their own colors
- [✓ Choose the color:]
  - visited link 🟥
  - unvisited link 🟦
- [✓ Underline links]

Figure 2. The Preferences Panel: the Color Tab.



Images:

- Load image automatically

Alterantives to visual...

- none
- [✓ preferred alternative:]
  - [✓ textual description]
  - audio
  - subtitles

Append or replace original content:

- append
- [✓ replace]

Figure 3. The Preferences Panel: the Visual Tab.

Through the Preferences Panel, the user can also choose and fix traditional browser preferences. This way, the user can have a more complete view of all the kind of customizations he/she could enjoy and can configure a wide set of characteristics, by using only one tool. Such a mechanism allows users to choose and set just once (for instance, during the browser installation) his/her preferences and needs, so as to enjoy adapted and accessible Web content anytime he/she access the browser. Obviously, through the Preference Panel, the user can change his/her preferences whenever he/she wants. Such new preferences will be immediately available. User's configurations are set through JavaScript and saved by the Preferences Storage System [28].

The content transcoding system of the Firefox extension prototype 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. 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 it and show it close to the related images. Moreover, also changes to the CSS are applied by using JavaScript. Some scripts can directly modify single CSS rules, while some others create a new CSS file, by implementing the user's preferences and needs, and then they substitute the old CSS with this new one in the HTML code of the Web page. With this kind of scripts it is possible to change the colors used in the Web page or the font family. Finally, also script transcoding is implemented by using JavaScript. Some of this kind of scripts are developed in order to automatically change Web 2.0 scripts (for instance to avoid automatic refreshing and updating of the Web page), while some others are designed to be performed only with a proper Web application, since it is not always possible to automatically identify and modify AJAX scripts in a feasible and effective way. This means that our prototype applies adequate scripts in order to transcode specific Web application pages (in particular when the users request Web 2.0 social networks content and services), as screen readers act with desktop applications.

5. USE CASE: IMPROVING FACEBOOK ACCESSIBILITY

In this Section we are going to describe the use of our prototype with the most famous and widespread Web 2.0 social network: Facebook. We will illustrate how we have designed and implemented the scripts which enhance Facebook accessibility and we will show results for some kinds of user profiles.

First of all, we have analyzed Facebook characteristics which affect its accessibility and we have identified main problems and their solutions. Such solutions have been applied by developing scripts for our browser extension GAPforAPE, in order to increase Facebook accessibility. Our analysis has begun with the study of how people with disabilities use assistive technologies to navigate the Web and, in particular, Facebook [8]. A discussion group of people with disabilities has been involved, answering to some interviews and participating in the design phase. A second group of users with disabilities has been engaged to test the system. Such testing phase is still an ongoing activity.

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 tool and the frequency they use it. 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;
- 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.

To improve Facebook accessibility we have designed and implemented a set of several scripts which face such problems and in particular they:

- Label text links in a correct way, so that none of them is ambiguous.
- 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 refresh the pages.
- 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 heavy the navigation with a screen reader.
- Provide a facilitated chat, on the basis of WAI-ARIA live regions roles (adequately added by our transcoding system).

When a blind user declares his/her preferences through the GAPforAPE Preferences Panel, he/she could choose to substitute images with textual alternatives or which is the preferred kind of alternative (textual, auditory, etc.). Figure 4 shows a screenshot of the original Facebook wall, while Figure 5 depicts the same wall with the application of GAPforAPE scripts.

Now let us consider users with low vision. They face different problems and it is very difficult to meet their needs, since there are several kinds and levels of this visual disability. The Preferences Panel allows users to set a wide group of configurations in a very detailed manner, so as to better adapt the Web pages and to better meet users' needs.

An accessible version of Facebook User's Profile page is shown in Figure 6: through our GAPforAPE 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, while black background and yellow texts provides a good level of contrast. Finally, Figure 7 shows another accessible version of the same Facebook User's Profile depicted in Figure 6. The scripts applied in this case increase the text size dimensions, without changing any color, and reorganized the Web page layout so as to provide a simpler and easier to navigate interface.

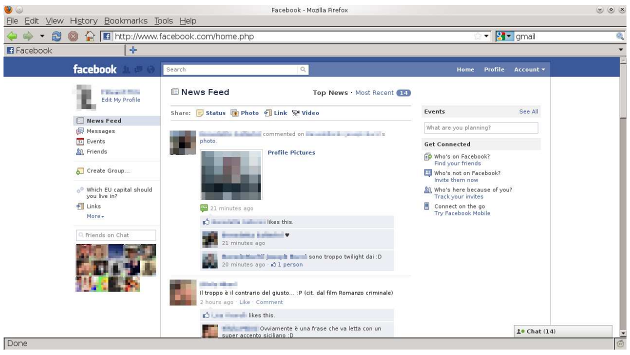


Figure 4. The Facebook wall.

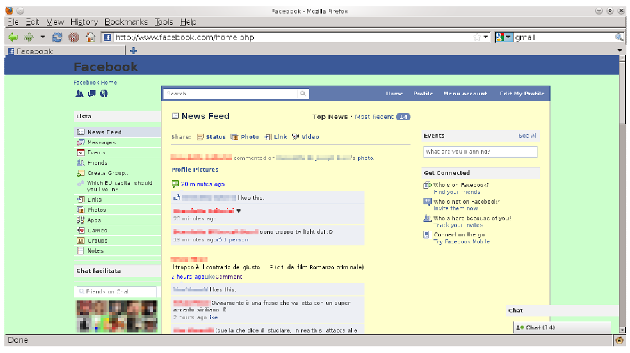
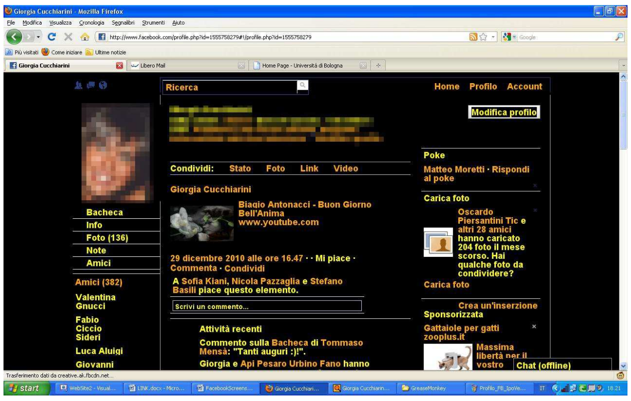

Facebook - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.facebook.com/home.php
Facebook
News Feed
Top News · Most Recent
Events
Get Connected
Chat facilitata
Chat
Done

Figure 5. The accessible Facebook wall.


Giorgia Cucchiarini - Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto
http://www.facebook.com/profile.php?id=1555758279#!/profile.php?id=1555758279
Più visitati Come iniziare Ultime notizie
Giorgia Cucchiarini
Libero Mail
Home Page - Università di Bologna
Ricerca
Home Profilo Account
Modifica profilo
Bacheca
Info
Foto (136)
Note
Amici
Condividi: Stato Foto Link Video
Poke
Carica foto
Crea un'inserzione Sponsorizzata
Chat (offline)
Trasferimento dati da creative.ak.fbcdn.net...
start

[redacted]

Giorgia Cucchiarini

Biagio Antonacci - Buon Giorno Bell'Anima www.youtube.com

29 dicembre 2010 alle ore 16.47 · · Mi piace · Commenta · Condividi

A Sofia Kiani, Nicola Pazzaglia e Stefano Basili piace questo elemento.

Scrivi un commento...

Attività recenti

Commento sulla Bacheca di Tommaso Mensà: "Tanti auguri :)!".

Giorgia e Api Pesaro Urbino Fano hanno

Amici (382)

Valentina Gnucci

Fabio Ciccio Sideri

Luca Aluigi

Giovanni

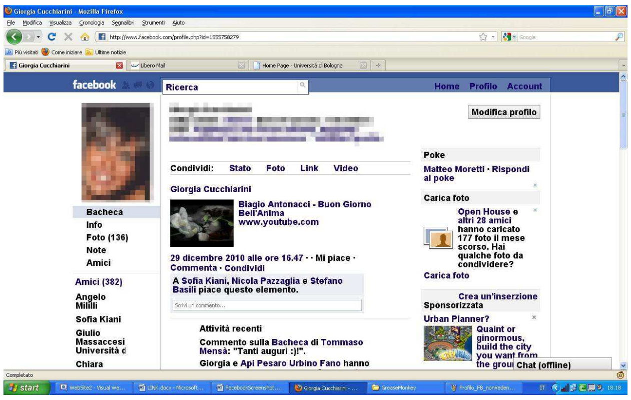
Matteo Moretti · Rispondi al poke

Oscardo Piersantini Tic e altri 28 amici hanno caricato 204 foto il mese scorso. Hai qualche foto da condividere?

Gattaiole per gatti zooplus.it

Massima libertà per il vostro

Figure 6. The Facebook user's profile adapted for low vision.


Giorgia Cucchiarini - Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto
http://www.facebook.com/profile.php?id=1555758279
Più visitati Come iniziare Ultime notizie
Giorgia Cucchiarini
Libero Mail
Home Page - Università di Bologna
facebook
Ricerca
Home Profilo Account
Modifica profilo
Bacheca
Info
Foto (136)
Note
Amici
Condividi: Stato Foto Link Video
Poke
Carica foto
Crea un'inserzione Sponsorizzata
Chat (offline)
Completato
start

[redacted]

Giorgia Cucchiarini

Biagio Antonacci - Buon Giorno Bell'Anima www.youtube.com

29 dicembre 2010 alle ore 16.47 · · Mi piace · Commenta · Condividi

A Sofia Kiani, Nicola Pazzaglia e Stefano Basili piace questo elemento.

Scrivi un commento...

Attività recenti

Commento sulla Bacheca di Tommaso Mensà: "Tanti auguri :)!".

Giorgia e Api Pesaro Urbino Fano hanno

Amici (382)

Angelo Mililli

Sofia Kiani

Giulio Massaccesi Università d

Chiara

Matteo Moretti · Rispondi al poke

Open House e altri 28 amici hanno caricato 177 foto il mese scorso. Hai qualche foto da condividere?

Urban Planner?

Quaint or ginormous, build the city you want from the grou

Figure 7. The Facebook user's profile adapted for low vision.

6. RELATED WORKS

Other works have been conducted with the aim of improving the accessibility of Web 2.0 applications and Web content developed with AJAX technologies. This section is devoted to describe a brief comparison between our proposed system and main related works in this context.

In [30] the authors present an improvement of SADIe, a system original developed to transform static Web pages (by manipulating the DOM of the pages) with the aim of providing content that can be better exploited by screen readers. The proposed SADIe improvement uses CSS annotations to describe the roles of Web page elements and to generate AxsJAX framework code and insert it into Web pages. Thus assistive technologies can interact with dynamic content, making it more accessible to users with disabilities.

The authors of [31] propose the adoption of WAI-ARIA into the Web-based eBuddy Instant Messaging platform (by using live regions). The paper describes two configurations of live regions:

A. Configuration A announces all updates with a polite priority for all marked live region DOM updates. Thus, all intended visual updates in the client were exposed to the user's assistive technology in the order they were received by the client. This solution is affected by the provision of many unrelated updates without any mechanism devoted to the identification of update priority.
B. Configuration B solves the problem of many unrelated updates by introducing a tally queue, with the aim of filtering updates.

Main similarities and differences between these projects and our proposed system are listed as follows:

- Our system applies content transcoding operations on the basis of a user's profile. This allows meeting users' preferences and needs, with the aim of maximizing their inclusion in enjoying Web 2.0 applications.
- Our system adapts Web content by transcoding the DOM page, the CSS rules and Web page scripts.
- Our system adapts CSS rules, but it does not consider CSS annotations in order to identify Web pages elements roles and then to adapt the content on the basis of them.
- Our system acts like a screen reader: when available, it applies specific scripts to transcode a specific application, otherwise it transcodes Web content on the basis of a default set of scripts.
- Our system operates on live regions in a way which is similar to the Configuration B proposed in [31].

7. CONCLUSION AND FUTURE WORKS

GAPforAPE is a system based on a set of Web browser extensions, with the aim of providing a content transcoding system which adapts Web pages in order to meet each user's preferences and needs, enhancing Web accessibility.

GAPforAPE exploits scripts at client-side to provide "the best Web content to each user": the same Web page is delivered to any user, but our Web browser extensions adapt it (transcoding Web content, modifying CSS rules, the HTML DOM and also Web pages scripts), so as to meet user's needs and preferences. The main idea is based on the GreaseMonkey extension and the AccessMonkey Framework, but our system is enriched with a user profiling system, which is based on the well-known standard IMS ACCLIP, in order to clearly and deeply describe user's needs and preferences with the aim of better adapting the Web content.

Another improvement introduced by our system is the identification of the requested Web application, so as to apply to its content the adequate scripts transcoding. This allows to effectively improving the accessibility of Web applications which are strongly based on Web 2.0 technologies, such as Facebook and other famous and widely used social networks. Indeed, it is not always possible to automatically identify and modify AJAX scripts in a unique, feasible and effective way. The idea of ad hoc scripts has been inspired by the behavior of well-known screen readers, such as Jaws.

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

Some people with disabilities have been invited to report how they use their assistive technologies/tools while navigating Facebook and which barriers they have to face during this activity. A second group of users with disabilities have been involved in the test of ad hoc scripts to surmount barriers on this Social Network. This test phase is still ongoing.

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.

8. ACKNOLEDGEMENTS

The authors wish to thank Ludovico Antonio Muratori and Giorgia Cucchiarini for their precious support.

9. REFERENCES

[1] AbilityNet Web Accessibility Team. State of the eNation web accessibility reports: Social Networking Websites. Available from: http://www.abilitynet.org.uk/docs/enation/2008SocialNetworkingSites.pdf, 2008.

[2] Bigham, J.P., and Ladner, R.E. AccessMonkey: A Collaborative Scripting Framework for Web Users and Developers. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A2007)* (Banff, Alberta, Canada, May 2007). ACM Press, New York, NY, 2007, 25-34.

[3] Bigham, J.P. AccessMonkey: enabling and sharing end user accessibility improvements. *ACM SIGACCESS Accessibility and Computing - ASSETS 2007*, No. 89 (Sep. 2007), 3-6.

[4] Bohman. P. Introduction to Web Accessibility. Available from: http://www.webaim.org/intro/, 2003.

[5] Codeidol. Thinking about GreaseMonkey – Accessibility. Available from: http://codeidol.com/internet/Greasemonkey/Accessibility, 2011.

[6] Colajanni, M. and Lancellotti, R.. System Architectures for Web Content Adaptation Services. In *IEEE Distributed Systems online*, Vol. 5, No. 5, IEEE Communications Society, May 2004.

[7] DELI: A Delivery Context Library For CC/PP and UAProf. Available from: http://delicon.sourceforge.net/, 2007.

[8] Facebook. Available from: http://www.facebook.com/, 2011.

[9] Garrett, J.J. AJAX: A new approach to Web applications. Available from: http://www.adaptivepath.com/ideas/essays/archives/000385.php, 2005.

[10] Greaspot. GreaseMonkey Firefox Extension. Available from: http://greasemonkey.mozdev.org/, 2011.

[11] Hailpern, J., Guarino-Reid, L., Boardman, R., Annam, S. Web 2.0: blind to an accessible new world. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)* (Madrid, Spain, April 2009). ACM Press, New York, NY, 2009, 821-830.

[12] Harrison, L., and Treviranus, J. Accessible E-Learning - Demystifying IMS Specifications. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN 2003)* (Phoenix, Arizona, USA, 2003). In A. Rossett (Ed.), 2000-2003.

[13] IMS Global Learning Consortium. IMS Learner Information Package Accessibility for LIP. Available from: http://www.imsglobal.org/specificationdownload.cfm, 2002.

[14] IMS Global Learning Consortium. IMS Learner Information Profile (LIP). Available from: http://www.imsglobal.org/specificationdownload.cfm, 2002.

[15] Isaías, P., Miranda, P., Pífano, S. Critical Success Factors for Web 2.0 --- A Reference Framework. In *Proceedings of the 3rd International Conference on Online Communities and Social Computing: Held as Part of HCI International 2009 (OCSC '09)* (San Diego, California, USA, Jul. 2009). Lecture Notes in Computer Science, Springer, Berlin, Germany, 354-363.

[16] Freedom Scientific. JAWS for windows. Available from: http://www.freedomscientific.com, 2011.

[17] Laakko, T., and Hiltunen, T. Adapting Web Content to Mobile User Agents. *IEEE Internet Computing*, Vol. 9, No. 2, IEEE Communications Society, March-April 2005, 46-53.

[18] Lembree, D. Accessible Twitter: Articles and Feedback. Available from: http://www.accessibletwitter.com/articles.php, 2011.

[19] Mirri, S., Salomoni, P., Roccetti, M., Gay, G.R. Beyond Standards: Unleashing Accessibility on a Learning Content Management System. Transactions on Edutainment V, LNCS 6530, Springer, Berlin, Germany, Feb. 2011.

[20] Nevile, L., Cooper, M, Health, A., Rothberg M., and Treviranus, J. Learner-centred Accessibility for Interoperable Web-based Educational Systems. In *Proceedings of the 14th International World Wide Web Conference (WWW '05)* (Japan, May 2005). ACM Press, New York, NY, 2005.

[21] Open Mobile Alliance (OMA). User Agent Profile v. 1.1 Approved Enabler. Available from: http://www.openmobilealliance.org/release_program/uap_v11.html, 2002.

[22] Pilgrim, M. GreaseMonkey Hacks: Tips & Tools for Remixing the Web with Firefox. O'Reilly Media, 2005.

[23] Salomoni, P., Mirri, S., Ferretti, S., Roccetti, M. A multimedia broker to support accessible and mobile learning through learning objects adaptation. ACM Transactions on Internet Technology (TOIT) , Vo. 8 No. 2, ACM Press, Feb. 2008, 9-23.

[24] Santos, O. C., and Boticario, J. G. Improving learners' satisfaction in specification-based scenarios with dynamic inclusive support. In *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies (ICALT '08)* (Santander, Spain, Jul. 2008). IEEE Press, New York, NY, 2008, 491-495.

[25] Sun Microsystem Inc. JSR 188: CC/PP Processing. Available from: http://www.jcp.org/en/jsr/detail?id=188, 2007.

[26] Userscripts.org. Power-ups for your browser: scripts tagged accessibility. Available from: http://userscripts.org/tags/accessibility, 2011.

[27] World Wide Web Consortium. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. Available from: http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115, 2004.

[28] Mozilla Developer Network. XML User Interface Language (XUL). Available from: https://developer.mozilla.org/En/XUL, 2011.

[29] Zajicek, M. Web 2.0: hype or happiness? In *Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A2007)* (Banff, Alberta, Canada, May 2007). ACM Press, New York, NY, 35-39.

[30] Lunn, D., Harper, S. and Bechhofer, S. Combining SADIe and AxsJAX to Improve the Accessibility of Web Content. In *Proceedings of the 2009 International cross-disciplinary conference on Web accessibility (W4A2009)* (Madrid, Spain, April 2009). ACM Press, New York, NY, 2009, 75-38.

[31] Thiessen, P., and Hockema, S. WAI-ARIA Live Regions: eBuddy IM as a Case Example. In *Proceedings of the 2009 International cross-disciplinary conference on Web accessibility (W4A2010)* (Railegh, North Carolina, USA, April 2010). ACM Press, New York, NY, 2010.